



TUGAS AKHIR - KI141502

**Konstruksi Sistem Informasi Administrasi
Tugas Akhir Berbasis Aliran Kerja untuk
Institut Teknologi Sepuluh Nopember
Menggunakan Kerangka Kerja Pemrograman
Laravel**

Ifan Iqbal
NRP 5111 100 098

Dosen Pembimbing
Umi Laili Yuhana, S.Kom., M.Sc.
Diana Purwitasari, S.Kom., M.Sc.

JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2016



FINAL PROJECT - KI141502

Construction of Administration Information System of Final Project Based Work Flow for the Sepuluh Nopember Institute of Technology Using Laravel Programming Framework

Ifan Iqbal
NRP 5111 100 098

Advisor
Umi Laili Yuhana, S.Kom., M.Sc.
Diana Purwitasari, S.Kom., M.Sc.

DEPARTMENT OF INFORMATICS
Faculty of Information Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2016

KATA PENGANTAR

Alhamdulillah, Puji syukur kepada Allah SWT yang telah melimpahkan rahmat dan anugerah-Nya sehingga penulis dapat menyelesaikan tugas akhir yang berjudul :

“Rancang Bangun Aplikasi Rekomendasi Produk Berdasarkan Kedekatan Pertemanan pada Twitter Konstruksi Sistem Informasi Administrasi Tugas Akhir Berbasis Aliran Kerja untuk Institut Teknologi Sepuluh Nopember Menggunakan Kerangka Kerja Pemrograman Laravel”

Proses pengerjaan Tugas Akhir ini merupakan suatu kesempatan yang sangat berharga bagi penulis. Dengan pengerjaan Tugas Akhir ini penulis bisa memperdalam, meningkatkan, serta menerapkan apa yang telah penulis dapatkan selama menempuh perkuliahan di Teknik Informatika ITS. Harapan dari penulis semoga apa yang tertulis di dalam buku Tugas Akhir ini dapat bermanfaat bagi pengembangan ilmu pengetahuan.

Terselesaikannya buku Tugas Akhir ini tak lepas dari dukungan dan bantuan semua pihak, oleh karena itu penulis ingin menyampaikan terimakasih kepada:

1. Allah SWT atas limpahan rahmat-Nya sehingga penulis dapat menyelesaikan Tugas Akhir ini dengan baik.
2. Kedua orang tua penulis, Bapak Sunarto dan Ibu R. Puji Emi Sundari yang selalu mendoakan dan memberikan dukungan kepada penulis.
3. Ibu Umi Laili Yuhana dan Ibu Diana Purwitasari selaku dosen pembimbing yang telah bersedia meluangkan waktu untuk memberikan bimbingan dan motivasi selama proses pengerjaan tugas akhir ini.
4. Bapak, Ibu dosen Jurusan Teknik Informatika ITS yang telah banyak memberikan ilmu dan bimbingan yang tak ternilai harganya bagi penulis.
5. Seluruh staf dan karyawan FTIf ITS yang banyak memberikan kelancaran administrasi akademik kepada penulis.

6. Rekan-rekan Pusat Pengembangan SI dan Pusat Data & Pelaporan LPTSI yang selama ini telah memberikan dukungan kepada penulis dalam penyelesaian tugas akhir ini.
7. Serta pihak-pihak lain yang tidak dapat disebutkan di sini yang telah banyak membantu penulis dalam penyusunan tugas akhir ini.

Kesempurnaan tentu masih jauh tercapai pada Tugas Akhir ini. Oleh karena itu, penulis mengharapkan saran dan kritik yang membangun sebagai bahan perbaikan selanjutnya.

Surabaya, Januari 2016

Ifan Iqbal

LEMBAR PENGESAHAN

**Konstruksi Sistem Informasi Administrasi Tugas Akhir
Berbasis Aliran Kerja untuk Institut Teknologi Sepuluh
Nopember Menggunakan Kerangka Kerja Pemrograman
Laravel**

TUGAS AKHIR

Diajukan Guna Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Bidang Studi Rekayasa Perangkat Lunak
Program Studi S-1 Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh :

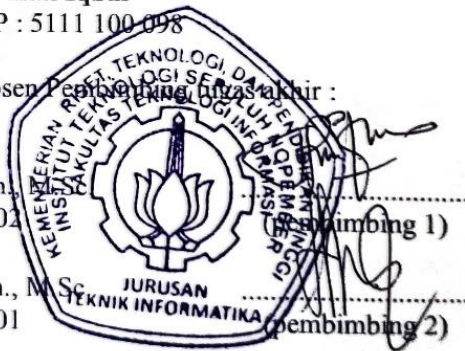
Ifan Iqbal

NRP : 5111 100 098

Disetujui oleh Dosen Pembimbing Tugas Akhir :

Umi Laili Yuhana, S.Kom., M.Sc.
NIP: 197906262005012002

Diana Purwitasari, S.Kom., M.Sc.
NIP: 197804102003122001



**SURABAYA
JANUARI 2016**

DAFTAR ISI

LEMBAR PENGESAHAN	vii
ABSTRAK	ix
ABSTRACT	xi
KATA PENGANTAR.....	xiii
DAFTAR ISI	xv
DAFTAR GAMBAR.....	xvii
DAFTAR TABEL	xxiii
DAFTAR KODE SUMBER.....	xxvii
BAB 1 PENDAHULUAN.....	29
1.1 Latar Belakang.....	29
1.2 Tujuan.....	30
1.3 Rumusan Permasalahan.....	30
1.4 Batasan Permasalahan	30
1.5 Metodologi	31
1.6 Sistematika Penulisan.....	32
BAB 2 DASAR TEORI.....	35
2.1 Prosedur Tugas Akhir.....	35
2.2 Organisasi dan Tata Kerja ITS	40
2.3 Konsep Aliran Kerja.....	40
2.4 Migrasi Sistem.....	41
2.5 Kerangka Kerja Pemrograman Laravel	41
BAB 3 ANALISIS DAN PERANCANGAN SISTEM.....	43
3.1 Analisis	43
3.1.1 Analisis Permasalahan.....	43
3.1.2 Deskripsi Umum Sistem	44
3.1.3 Spesifikasi Kebutuhan Perangkat Lunak	45
3.1.4 Aktor.....	46
3.1.5 Kasus Penggunaan.....	46
3.2 Perancangan Sistem.....	98
3.2.1 Perancangan Arsitektur Perangkat Lunak	99
3.2.2 Perancangan Proses Aplikasi.....	100
3.2.3 Perancangan Basis Data.....	106

3.2.4	Perancangan Antarmuka Pengguna	118
BAB 4	IMPLEMENTASI	133
4.1	Lingkungan Implementasi	133
4.1.1	Lingkungan Implementasi Perangkat Keras	133
4.1.2	Lingkungan Implementasi Perangkat Lunak	133
4.2	Implementasi Proses Aplikasi	133
4.2.1	Implementasi Proses Sistem Generik	134
4.2.2	Implementasi Proses Pengaturan Aliran Kerja	139
4.2.3	Implementasi Proses Pengaturan Parameter	144
4.3	Implementasi Antarmuka Pengguna	145
4.4	Implementasi Basis Data	146
4.4.1	Implementasi Struktur Basis Data	146
4.4.2	Implementasi Model	146
BAB 5	PENGUJIAN DAN EVALUASI	149
5.1	Lingkungan Pengujian	149
5.2	Skenario Pengujian	149
5.2.1	Pengujian Fungsionalitas	149
5.3	Evaluasi Pengujian Fungsionalitas	183
5.3.1	Evaluasi Pengujian Fungsionalitas	183
BAB 6	KESIMPULAN DAN SARAN	185
6.1	Kesimpulan	185
6.2	Saran	185
DAFTAR PUSTAKA		187
LAMPIRAN A. IMPLEMENTASI APLIKASI		189
BIODATA PENULIS		217

DAFTAR GAMBAR

Gambar 2.1 Prosedur Tugas Akhir Jurusan Teknik Informatika - FTIf.....	36
Gambar 2.2 Prosedur Tugas Akhir Jurusan Teknik Mesin - FTI	37
Gambar 2.3 Prosedur Tugas Akhir Jurusan Kimia - FMIPA	37
Gambar 2.4 Prosedur Tugas Akhir Jurusan Teknik Sistem Perkapalan – FTK.....	38
Gambar 2.5 Prosedur Tugas Akhir Jurusan Teknik Geomatika - FTSP.....	39
Gambar 2.6 Spesifikasi Aliran Kerja.....	41
Gambar 3.1 Gambaran Umum Perangkat Lunak	45
Gambar 3.2 Kasus Penggunaan Aktor Mahasiswa.....	48
Gambar 3.3 Kasus Penggunaan Aktor Dosen	48
Gambar 3.4 Kasus Penggunaan Aktor Koordinator	49
Gambar 3.5 Kasus Penggunaan Aktor Administrator	49
Gambar 3.6 Kasus Penggunaan Aktor Administrator Pusat.....	50
Gambar 3.7 Diagram Aktivitas Memilih Dosen Pembimbing	52
Gambar 3.8 Diagram Aktivitas Mengunggah Berkas Proposal...	54
Gambar 3.9 Diagram Aktivitas Melihat Progres Bimbingan dan Progres Buku	56
Gambar 3.10 Diagram Aktivitas Menambahkan Progres Bimbingan	57
Gambar 3.11 Diagram Aktivitas Menambahkan Progres Buku ..	59
Gambar 3.12 Diagram Aktivitas Memperbarui Progres Bimbingan	61
Gambar 3.13 Diagram Aktivitas Memperbarui Progres Buku	63
Gambar 3.14 Diagram Aktivitas Mengapus Progres Bimbingan	65
Gambar 3.15 Diagram Aktivitas Mengapus Progres Buku	66
Gambar 3.16 Diagram Aktivitas Melihat Jadwal Sidang Pribadi (Mahasiswa)	68
Gambar 3.17 Diagram Aktivitas Melihat Laporan Pemantauan Progres Mahasiswa.....	69
Gambar 3.18 Diagram Aktivitas Melakukan Verifikasi Progres.	72
Gambar 3.19 Diagram Aktivitas Melihat Jadwal Sidang Pribadi (dosen).....	73

Gambar 3.20 Diagram Aktivitas Melihat Halaman Kelola Sidang	74
Gambar 3.21 Diagram Aktivitas Menambahkan Jadwal Sidang.....	76
Gambar 3.22 Diagram Aktivitas Mengubah Data Jadwal Sidang	78
Gambar 3.23 Diagram Aktivitas Menghapus Jadwal Sidang.....	79
Gambar 3.24 Diagram Aktivitas Melihat Laporan-Laporan	81
Gambar 3.25 Diagram Aktivitas Melihat Daftar Mahasiswa	82
Gambar 3.26 Diagram Aktivitas Menambahkan Mahasiswa	84
Gambar 3.27 Diagram Aktivitas Mengubah Data Mahasiswa	86
Gambar 3.28 Diagram Aktivitas Menghapus Mahasiswa	87
Gambar 3.29 Diagram Aktivitas Melihat Daftar Nilai	88
Gambar 3.30 Diagram Aktivitas Memasukkan Nilai Tugas Akhir	90
Gambar 3.31 Diagram Aktivitas Melihat Halaman Pengelolaan Jurusan.....	91
Gambar 3.32 Diagram Aktivitas Melihat Halaman Pengelolaan Aliran Kerja.....	93
Gambar 3.33 Diagram Aktivitas Melihat Halaman Pengelolaan Hak Akses.....	94
Gambar 3.34 Diagram Aktivitas Melihat Halaman Pengelolaan Parameter.....	95
Gambar 3.35 Diagram Aktivitas Melihat Halaman Pengelolaan Menu.....	97
Gambar 3.36 Diagram Aktivitas Melihat Halaman Pengelolaan Peran.....	98
Gambar 3.37 Arsitektur Perangkat Lunak	99
Gambar 3.38 Aliran Kerja Utama SIMTA ITS	101
Gambar 3.39 Pemaketan Aliran Kerja Prosedur Tugas Akhir Bagi Pengguna Mahasiswa	105
Gambar 3.40 Model Data Fisik	108
Gambar 3.41 Antarmuka Kerangka Tata Letak.....	119
Gambar 3.42 Rancangan Antarmuka Halaman Login.....	119
Gambar 3.43 Rancangan Antarmuka Pemilihan Dosen Pembimbing.....	120

Gambar 3.44 Rancangan Antarmuka Halaman Unggah Proposal	121
Gambar 3.45 Rancangan Antarmuka Halaman Pengelolaan Progres.....	122
Gambar 3.46 Rancangan Antarmuka Halaman Penambahan Progres Bimbingan	123
Gambar 3.47 Rancangan Antarmuka Halaman Penambahan Progres Buku	124
Gambar 3.48 Rancangan Antarmuka Halaman Jadwal Sidang Pribadi Mahasiswa.....	125
Gambar 3.49 Rancangan Antarmuka Halaman Verifikasi Progres	125
Gambar 3.50 Rancangan Antarmuka Halaman Jadwal Sidang Pribadi Dosen	126
Gambar 3.51 Rancangan Antarmuka Halaman Kelola Sidang ..	126
Gambar 3.52 Rancangan Antarmuka Halaman Penambahan Sidang	127
Gambar 3.53 Rancangan Antarmuka Halaman Pengubahan Sidang	128
Gambar 3.54 Rancangan Antarmuka Halaman Kelola Data Mahasiswa	128
Gambar 3.55 Rancangan Antarmuka Halaman Penambahan Mahasiswa	129
Gambar 3.56 Rancangan Antarmuka Halaman Pengubahan Mahasiswa	130
Gambar 3.57 Rancangan Antarmuka Halaman Pengelolaan Nilai	131
Gambar 3.58 Rancangan Antarmuka Halaman Pemasukan Nilai	131
Gambar 5.1 Hasil Pengujian Fitur Memilih Dosen Pembimbing	151
Gambar 5.2 Hasil Pengujian Fitur Mengunggah Berkas Proposal	152
Gambar 5.3 Hasil Pengujian Fitur Penambahan Progres Bimbingan	154

Gambar 5.4 Hasil Pengujian Fitur Pengubahan Progres Bimbingan	155
Gambar 5.5 Hasil Pengujian Fitur Melihat Daftar Progres Bimbingan	155
Gambar 5.6 Hasil Pengujian Fitur Penambahan Progres Buku ..	156
Gambar 5.7 Hasil Pengujian Fitur Pengubahan Progres Buku ..	157
Gambar 5.8 Hasil Pengujian Fitur Melihat Daftar Progres Buku	157
Gambar 5.9 Hasil Pengujian Fitur Melihat Jadwal Sidang Pribadi (Mahasiswa)	158
Gambar 5.10 Hasil Pengujian Fitur Melihat Laporan Pemantauan Progres Mahasiswa	159
Gambar 5.11 Hasil Pengujian Fitur Melakukan Verifikasi Progres	160
Gambar 5.12 Hasil Pengujian Fitur Melihat Jadwal Sidang Pribadi (Dosen)	161
Gambar 5.13 Hasil Pengujian Fitur Penambahan Jadwal Sidang	163
Gambar 5.14 Hasil Pengujian Fitur Pengubahan Jadwal Sidang	164
Gambar 5.15 Hasil Pengujian Fitur Mengelola Jadwal Sidang ..	164
Gambar 5.16 Hasil Pengujian Fitur Melihat Laporan-Laporan ..	165
Gambar 5.17 Hasil Pengujian Fitur Melihat Laporan Monitoring Progres Seluruh Mahasiswa	165
Gambar 5.18 Hasil Pengujian Fitur Penambahan Data Mahasiswa	167
Gambar 5.19 Hasil Pengujian Fitur Melihat Halaman Kelola Mahasiswa	167
Gambar 5.20 Hasil Pengujian Fitur Mengelola Nilai Tugas Akhir	169
Gambar 5.21 Hasil Pengujian Fitur Mengisi Nilai Tugas Akhir	169
Gambar 5.22 Hasil Pengujian Fitur Mengelola Jurusan	171
Gambar 5.23 Hasil Pengujian Fitur Mengelola Aliran Kerja	172
Gambar 5.24 Hasil Pengujian Fitur Mengelola Hak Akses	174

Gambar 5.25 Hasil Pengujian Fitur Mengelola Parameter	176
Gambar 5.26 Hasil Pengujian Fitur Mengelola Menu.....	177
Gambar 5.27 Hasil Pengujian Fitur Mengelola Peran.....	180
Gambar 5.28 Hasil Pengujian Penambahan Jurusan	182
Gambar 5.29 Hasil Pengujian Pemetaan Proses Aliran Kerja...	182

DAFTAR TABEL

Tabel 3.1 Daftar Kasus Penggunaan.....	47
Tabel 3.2 Spesifikasi Kasus Penggunaan Memilih Dosen Pembimbing.....	51
Tabel 3.3 Spesifikasi Kasus Penggunaan Mengunggah Berkas Proposal.....	52
Tabel 3.4 Spesifikasi Kasus Penggunaan Melihat Progres Bimbingan Dan Progres Buku.....	55
Tabel 3.5 Spesifikasi Kasus Penggunaan Menambahkan Progres Bimbingan	56
Tabel 3.6 Spesifikasi Kasus Penggunaan Menambahkan Progres Buku	57
Tabel 3.7 Spesifikasi Kasus Penggunaan Memperbarui Progres Bimbingan	59
Tabel 3.8 Spesifikasi Kasus Penggunaan Meperbarui Progres Buku	61
Tabel 3.9 Spesifikasi Kasus Penggunaan Menghapus Progres Bimbingan	63
Tabel 3.10 Spesifikasi Kasus Penggunaan Menghapus Progres Buku	65
Tabel 3.11 Spesifikasi Kasus Penggunaan Melihat Jadwal Sidang Pribadi (Mahasiswa).....	67
Tabel 3.12 Spesifikasi Kasus Penggunaan Melihat Laporan Pemantauan Progres Mahasiswa	68
Tabel 3.13 Spesifikasi Kasus Penggunaan Melakukan Verifikasi Progres.....	71
Tabel 3.14 Spesifikasi Kasus Penggunaan Melihat Jadwal Sidang Pribadi (Dosen).....	72
Tabel 3.15 Spesifikasi Kasus Penggunaan Melihat Halaman Kelola Sidang.....	74
Tabel 3.16 Spesifikasi Kasus Penggunaan Menambahkan Jadwal Sidang.....	75
Tabel 3.17 Spesifikasi Kasus Penggunaan Mengubah Data Jadwal Sidang.....	76

Tabel 3.18 Spesifikasi Kasus Penggunaan Menghapus Jadwal Sidang.....	78
Tabel 3.19 Spesifikasi Kasus Penggunaan Melihat Laporan-Laporan.....	80
Tabel 3.20 Spesifikasi Kasus Penggunaan Melihat Daftar Mahasiswa.....	81
Tabel 3.21 Spesifikasi Kasus Penggunaan Menambahkan Mahasiswa.....	82
Tabel 3.22 Spesifikasi Kasus Penggunaan Mengubah Data Mahasiswa.....	84
Tabel 3.23. Spesifikasi Kasus Penggunaan Menghapus Mahasiswa.....	86
Tabel 3.24 Spesifikasi Kasus Penggunaan Melihat Daftar Nilai.....	88
Tabel 3.25 Spesifikasi Kasus Penggunaan Memasukkan Nilai Tugas Akhir.....	89
Tabel 3.26 Spesifikasi Kasus Penggunaan Melihat Halaman Pengelolaan Jurusan.....	90
Tabel 3.27 Spesifikasi Kasus Penggunaan Melihat Halaman Pengelolaan Aliran Kerja.....	92
Tabel 3.28 Spesifikasi Kasus Penggunaan Melihat Halaman Pengelolaan Hak Akses.....	93
Tabel 3.29 Spesifikasi Kasus Penggunaan Melihat Halaman Pengelolaan Parameter.....	94
Tabel 3.30 Spesifikasi Kasus Penggunaan Melihat Halaman Pengelolaan Menu.....	96
Tabel 3.31 Spesifikasi Kasus Penggunaan Melihat Halaman Pengelolaan Peran.....	97
Tabel 3.32 Rancangan Tabel <i>Chapters</i>	107
Tabel 3.33 Rancangan Tabel <i>Book_Progresses</i>	109
Tabel 3.34. Rancangan Tabel <i>Interest Areas</i>	109
Tabel 3.35 Rancangan Tabel <i>Education Degrees</i>	109
Tabel 3.36 Rancangan Tabel <i>Degree Types</i>	110
Tabel 3.37 Rancangan Tabel <i>Meeting Types</i>	110
Tabel 3.38 Rancangan Tabel <i>Rooms</i>	110
Tabel 3.39 Rancangan Tabel <i>Majors</i>	110

Tabel 3.40 Rancangan Tabel <i>External Lecturers</i>	111
Tabel 3.41 Rancangan Tabel <i>Meetings</i>	111
Tabel 3.42 Rancangan Tabel <i>Progress Statuses</i>	111
Tabel 3.43 Rancangan Tabel <i>Projects</i>	112
Tabel 3.44 Rancangan Tabel <i>Proposals</i>	112
Tabel 3.45 Rancangan Tabel <i>Lecturer Project</i>	113
Tabel 3.46 Rancangan Tabel <i>Lecturers</i>	113
Tabel 3.47 Rancangan Tabel <i>Students</i>	113
Tabel 3.48 Rancangan Tabel <i>Lecturer Meeting</i>	114
Tabel 3.49 Rancangan Tabel <i>Progresses</i>	114
Tabel 3.50 Rancangan Tabel <i>Roles</i>	115
Tabel 3.51 Rancangan Tabel <i>Permissions</i>	115
Tabel 3.52 Rancangan Tabel <i>Users</i>	115
Tabel 3.53 Rancangan Tabel <i>Permission Role</i>	116
Tabel 3.54 Rancangan Tabel <i>Role User</i>	116
Tabel 3.55 Rancangan Tabel <i>Workflows</i>	117
Tabel 3.56 Rancangan Tabel <i>Menus</i>	117
Tabel 3.57 Rancangan Tabel <i>Menu Role</i>	117
Tabel 3.58 Rancangan Tabel <i>Parameters</i>	117
Tabel 3.59 Rancangan Tabel <i>Major Parameter</i>	118
Tabel 5.1 Pengujian Fitur Memilih Dosen Pembimbing	150
Tabel 5.2 Pengujian Fitur Mengunggah Berkas Proposal	151
Tabel 5.3 Pengujian Fitur Mengelola Progres Bimbingan dan Progres Buku	153
Tabel 5.4 Pengujian Fitur Melihat Jadwal Sidang Pribadi (Mahasiswa)	158
Tabel 5.5 Pengujian Fitur Melihat Laporan Pemantauan Progres Mahasiswa	159
Tabel 5.6 Pengujian Fitur Melakukan Verifikasi Progres	160
Tabel 5.7 Pengujian Fitur Melihat Jadwal Sidang Pribadi (Dosen)	161
Tabel 5.8 Pengujian Fitur Mengelola Jadwal Sidang	162
Tabel 5.9 Pengujian Fitur Melihat Laporan-Laporan	165
Tabel 5.10 Pengujian Fitur Mengelola Data Mahasiswa.....	166
Tabel 5.11 Pengujian Fitur Mengelola Nilai Tugas Akhir	168

Tabel 5.12 Pengujian Fitur Mengelola Jurusan	169
Tabel 5.13 Pengujian Fitur Mengelola Aliran Kerja	171
Tabel 5.14 Pengujian Fitur Mengelola Hak Akses	173
Tabel 5.15 Pengujian Fitur Mengelola Parameter	175
Tabel 5.16 Pengujian Fitur Mengelola Menu	176
Tabel 5.17 Pengujian Fitur Mengelola Peran	179
Tabel 5.18 Pengujian Aliran Kerja Jurusan	180
Tabel 5.19 Rangkuman Hasil Pengujian Fungsionalitas	183

DAFTAR KODE SUMBER

Kode Sumber 4.1 Pemilihan Dosen Pembimbing	134
Kode Sumber 4.2 Pengunggahan Proposal.....	135
Kode Sumber 4.3 Progres Bimbingan	136
Kode Sumber 4.4 Verifikasi Progres Bimbingan	137
Kode Sumber 4.5 Jadwal Sidang	138
Kode Sumber 4.6 Pengelolaan Nilai.....	139
Kode Sumber 4.7 Pengelolaan Jurusan	140
Kode Sumber 4.8 Pengelolaan Menu	141
Kode Sumber 4.9 Pemilihan Pengelolaan Hak Akses	142
Kode Sumber 4.10 Pemilihan Pengelolaan Peran	142
Kode Sumber 4.11 Pengelolaan Aliran Kerja	144
Kode Sumber 4.12 Pengaturan Parameter	145
Kode Sumber A.1 Migrasi Basis Data	197
Kode Sumber A.2 Model BookProgress	198
Kode Sumber A.3 Model Chapter	199
Kode Sumber A.4 Model DegreeType	199
Kode Sumber A.5 Model EducationDegree	200
Kode Sumber A.6 Model ExternalLecturer.....	200
Kode Sumber A.7 Model InterestArea	201
Kode Sumber A.8 Model Lecturer	202
Kode Sumber A.9 Model Major	202
Kode Sumber A.10 Model Meeting	204
Kode Sumber A.11 Model MeetingSession	204
Kode Sumber A.12 Model MeetingType	205
Kode Sumber A.13 Model Menu	206
Kode Sumber A.14 Model Parameter	206
Kode Sumber A.15 Model Progress	207
Kode Sumber A.16 Model ProgressStatus	208
Kode Sumber A.17 Model.....	209
Kode Sumber A.18 Model Proposal.....	210
Kode Sumber A.19 Model Room	211
Kode Sumber A.20 Model Student	212
Kode Sumber A.21 Model User	214

Kode Sumber A.22 Model Workflow214

Kode Sumber A.23 Model Permission.....215

Kode Sumber A.24 Model Role215

BAB 1

PENDAHULUAN

Pada bab ini akan dipaparkan mengenai garis besar tugas akhir yang meliputi latar belakang, tujuan, rumusan dan batasan permasalahan, metodologi pembuatan tugas akhir, dan sistematika penulisan.

1.1 Latar Belakang

Mengelola dan memantau proses pengerjaan tugas akhir mahasiswa merupakan salah satu kewajiban setiap jurusan di Institut Teknologi Sepuluh Nopember (ITS). Berdasarkan praktik di lapangan, kegiatan tersebut dilaksanakan oleh dosen dan petugas jurusan yang ditunjuk. Setiap jurusan memiliki cara yang berbeda-beda dalam mengelola dan memantau proses pengerjaan tugas akhir mahasiswa tersebut. Dalam pengelolaan dan pemantauan tersebut, ada jurusan yang memanfaatkan sistem informasi terintegrasi dalam jaringan dan ada pula yang tidak, yaitu di mana semua proses masih dilakukan secara manual dan di luar jaringan.

Pemanfaatan sistem informasi dalam mengelola dan memantau tugas akhir mahasiswa dapat memberikan keuntungan kepada jurusan pengguna. Namun, usaha setiap jurusan untuk mengembangkan sistem informasinya masing-masing dapat terhambat oleh kendala tersendiri di masa depan, seperti perubahan kebijakan yang mempengaruhi proses pengelolaan dan pemantauan tugas akhir di semua jurusan di ITS. Oleh karena itu, dibutuhkan sebuah sistem informasi administrasi tugas akhir yang bersifat generik yang dapat memenuhi kebutuhan setiap jurusan di ITS.

Sistem informasi administrasi tugas akhir yang bersifat generik merupakan sistem informasi yang merangkum berbagai proses bisnis, dari berbagai jurusan di ITS, menjadi sebuah sistem tunggal dengan struktur data atau basis data yang sama. Setiap jurusan dapat menjalankan proses bisnisnya masing-masing di dalam sistem generik tersebut tanpa mempengaruhi proses bisnis yang dijalankan oleh jurusan lainnya.

Konstruksi sistem informasi administrasi tugas akhir dalam tugas akhir ini diharapkan dapat menjadi solusi yang dapat

memenuhi kebutuhan setiap jurusan di ITS dalam melaksanakan proses pengelolaan dan pemantauan proses pengerjaan tugas akhir mahasiswa. Jika setiap jurusan menggunakan satu sistem yang sama, ITS akan mendapatkan keuntungan dari segi biaya, waktu, dan kemudahan akses data yang terpusat dan seragam.

1.2 Tujuan

Tujuan pembuatan tugas akhir ini adalah sebagai berikut.

1. Membuat sistem informasi administrasi tugas akhir yang bersifat generik untuk ITS.
2. Membuat sistem informasi administrasi tugas akhir yang dapat digunakan oleh berbagai jenis pengguna di berbagai jurusan di ITS.
3. Membuat sistem informasi administrasi tugas akhir yang dapat disesuaikan terhadap kebutuhan berbagai jurusan di ITS.

1.3 Rumusan Permasalahan

Rumusan masalah yang diangkat dalam tugas akhir ini adalah sebagai berikut.

1. Bagaimana membuat sistem informasi administrasi tugas akhir yang bersifat generik untuk ITS?
2. Bagaimana eksekusi aliran kerja sistem yang digunakan oleh berbagai jurusan di ITS dengan jenis pengguna dan hak akses yang juga beragam?
3. Apa saja parameter yang dapat diubah dan disesuaikan terhadap kebutuhan berbagai jurusan di ITS?
4. Bagaimana mengimplementasikan sistem informasi administrasi tugas akhir dengan menggunakan kerangka kerja pemrograman Laravel?

1.4 Batasan Permasalahan

Permasalahan yang akan diselesaikan dalam tugas akhir ini memiliki batasan-batasan sebagai berikut.

1. Sistem informasi administrasi tugas akhir merupakan aplikasi berbasis web dengan kerangka kerja pemrograman Laravel.
2. Manajemen tugas akhir dalam sistem dibatasi hanya untuk mahasiswa tahap sarjana saja.

3. Proses bisnis yang menjadi sumber kebutuhan utama dalam pembangunan sistem informasi generik berasal dari tiga jurusan yang berbeda.
4. Jurusan yang dijadikan sebagai jurusan pengguna dalam studi kasus adalah jurusan Teknik Mesin dan jurusan Teknik Informatika.
5. Sistem tidak terintegrasi dengan Sistem Informasi Akademik ITS.

1.5 Metodologi

Langkah-langkah yang ditempuh dalam pengerjaan tugas akhir ini yaitu:

1. Studi Literatur

Pada tahap ini dilakukan pengumpulan informasi mengenai kerangka kerja pemrograman Laravel beserta pustaka pendukungnya, konsep aliran kerja, hingga struktur organisasi di ITS. Informasi dan pengetahuan hasil mempelajari hal tersebut dibutuhkan dalam proses perancangan dan implementasi sistem yang dibangun. Adapun rincian literatur yang digunakan adalah sebagai berikut.

 1. Organisasi dan Tata Kerja ITS.
 2. Konsep Aliran Kerja.
 3. Migrasi Sistem.
 4. Kerangka Kerja Pemrograman Laravel.
2. Analisis dan Perancangan Sistem

Pada tahap ini dilakukan analisis dan pendefinisian kebutuhan sistem untuk masalah yang sedang dihadapi. Perancangan sistem dilakukan dalam beberapa tahap, yaitu:

 - a. analisis kasus pengguna;
 - b. perancangan proses aplikasi;
 - c. perancangan antar muka; dan
 - d. perancangan basis data.
3. Implementasi

Pada tahap ini dilakukan pembuatan elemen perangkat lunak. Sistem yang dibuat berpedoman pada rancangan yang telah dibuat pada proses perancangan dan analisis sistem.
4. Pengujian dan Evaluasi

Pada tahap ini dilakukan pengujian terhadap elemen perangkat lunak. Pengujian dan evaluasi perangkat dilakukan untuk mengevaluasi hasil analisis program.

5. Penyusunan Buku Tugas Akhir

Pada tahap ini dilakukan dokumentasi dan pelaporan dari seluruh konsep, dasar teori, implementasi, proses yang telah dilakukan, dan hasil-hasil yang telah didapatkan selama pengerjaan tugas akhir.

1.6 Sistematika Penulisan

Penulisan buku tugas akhir ini bertujuan untuk mendapatkan gambaran dari pengerjaan tugas akhir ini. Selain itu, diharapkan dapat berguna untuk pembaca yang tertarik untuk melakukan pengembangan lebih lanjut. Secara garis besar, buku tugas akhir terdiri atas beberapa bagian seperti berikut ini.

Bab I Pendahuluan

Bab ini berisi latar belakang masalah, tujuan dan manfaat pembuatan tugas akhir, permasalahan, batasan masalah, metodologi yang digunakan, dan sistematika penyusunan tugas akhir.

Bab II Dasar Teori

Bab ini membahas beberapa teori penunjang yang berhubungan dengan pokok pembahasan dan mendasari pembuatan tugas akhir ini.

Bab III Analisis dan Perancangan Sistem

Bab ini membahas mengenai perancangan perangkat lunak. Perancangan perangkat lunak meliputi perancangan data, arsitektur, proses dan perancangan antarmuka pada kakas.

Bab IV Implementasi

Bab ini berisi implementasi dari perancangan perangkat lunak.

Bab V Pengujian dan Evaluasi

Bab ini membahas pengujian fungsionalitas dengan metode pengujian *black box* untuk mengetahui kesesuaian hasil keluaran sistem. dan untuk mengukur performa sistem dilakukan perhitungan tingkat presisi dan

recall.

Bab VI Kesimpulan

Bab ini berisi kesimpulan dari hasil pengujian yang dilakukan. Bab ini membahas saran-saran untuk pengembangan sistem lebih lanjut.

Daftar Pustaka

Merupakan daftar referensi yang digunakan untuk mengembangkan tugas akhir.

Lampiran

Merupakan bab tambahan yang berisi daftar istilah yang penting pada aplikasi ini.

[Halaman ini sengaja dikosongkan]

BAB 2

DASAR TEORI

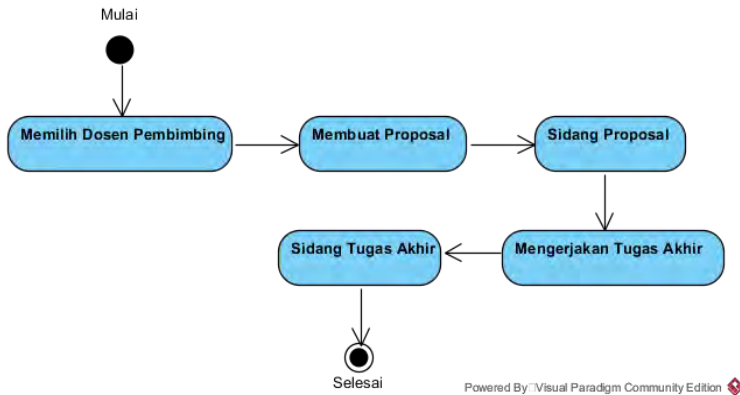
Pada bab ini akan dibahas mengenai teori-teori yang menjadi dasar dari pembuatan tugas akhir. Teori-teori tersebut meliputi tugas akhir, organisasi dan tata kerja ITS, konsep aliran kerja, migrasi sistem, dan kerangka kerja pemrograman Laravel.

2.1 Prosedur Tugas Akhir

Tugas akhir merupakan akhir dari proses pendidikan untuk program studi S1 yang harus diselesaikan mahasiswa [1]. Penyelesaian tugas akhir ditujukan untuk memberikan pengalaman bagi mahasiswa dalam menyusun karya ilmiah yang berkaitan dengan penyelesaian terhadap suatu pokok permasalahan sesuai dengan bidang keahlian yang telah dipilih mahasiswa. Tugas akhir bersifat komprehensif, ditulis sesuai dengan norma-norma ilmiah dan harus dipertanggungjawabkan baik dalam forum seminar tugas akhir yang bersifat umum dan terbuka maupun ujian lisan yang bersifat tertutup di depan tim penguji.

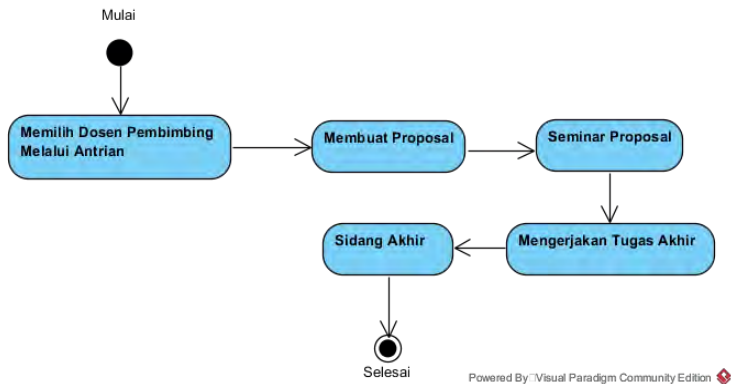
Di ITS, penyelenggaraan tugas akhir memiliki proses bisnis yang berbeda-beda untuk tiap jurusan. ITS belum memiliki kebijakan yang mengatur prosedur tugas akhir. Hal tersebut memunculkan berbagai perbedaan dalam pelaksanaan prosedur tugas akhir pada jurusan-jurusan di ITS.

Prosedur tugas akhir merupakan tahap kegiatan yang harus ditempuh oleh mahasiswa untuk menyelesaikan tugas akhir. Mahasiswa mengikuti prosedur tugas akhir sesuai yang ditetapkan oleh jurusannya masing-masing. Di ITS, prosedur tugas akhir berbeda-beda untuk tiap jurusan. Berikut ini merupakan beberapa prosedur tugas akhir dari beberapa jurusan di ITS.



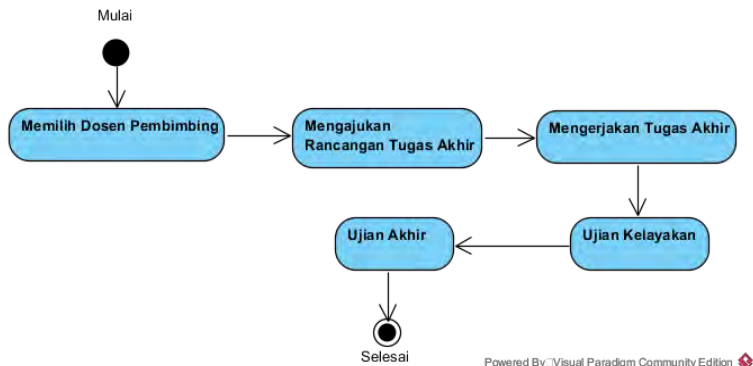
Gambar 2.1 Prosedur Tugas Akhir Jurusan Teknik Informatika - FTIF

Prosedur tugas akhir Jurusan Teknik Informatika ditunjukkan pada Gambar 2.1. Mahasiswa berhak memilih dua orang dosen pembimbing. Setelah itu, mahasiswa membuat proposal tugas akhir dilanjutkan dengan sidang proposal. Sidang proposal dihadiri oleh mahasiswa, dosen pembimbing dan dosen penguji. Apabila proposal diterima, mahasiswa dipersilakan untuk memulai pengerjaan tugas akhir. Apabila tugas akhir telah siap, sidang tugas akhir bisa dilaksanakan. Sidang tugas akhir memiliki format yang mirip dengan sidang proposal, hanya saja yang menjadi pembahasan adalah tugas akhir mahasiswa. Sidang proposal dan tugas akhir bersifat tertutup, tidak dibuka untuk publik.



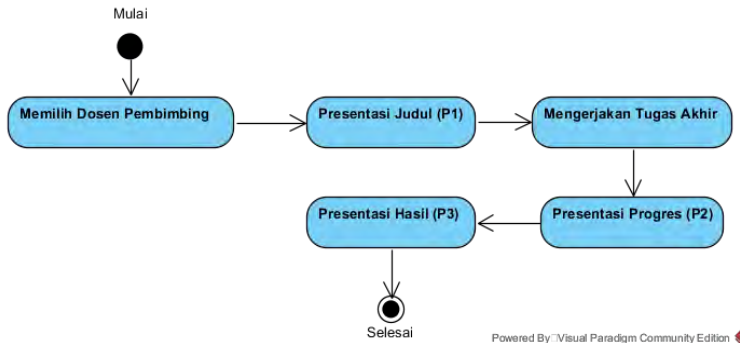
Gambar 2.2 Prosedur Tugas Akhir Jurusan Teknik Mesin - FTI

Prosedur tugas akhir Jurusan Teknik Mesin ditunjukkan pada Gambar 2.2. Mahasiswa memilih seorang dosen pembimbing melalui proses antrian dosen pembimbing. Jurusan Teknik Mesin mengatur kuota mahasiswa bimbingan untuk setiap dosen secara ketat. Setelah mendapatkan dosen pembimbing, mahasiswa mempertanggungjawabkan proposalnya dalam seminar proposal. Apabila lulus dari seminar proposal, mahasiswa dapat mengerjakan tugas akhir hingga selesai. Setelah siap, tugas akhir dapat disidangkan dalam sidang akhir.



Gambar 2.3 Prosedur Tugas Akhir Jurusan Kimia - FMIPA

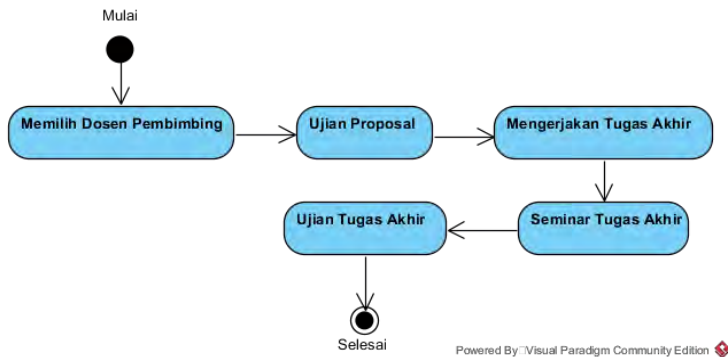
Prosedur tugas akhir Jurusan Kimia ditunjukkan pada Gambar 2.3. Mula-mula, mahasiswa memilih seorang dosen pembimbing. Setelah mendapatkan dosen pembimbing, mahasiswa diwajibkan membuat rancangan tugas akhir. Rancangan tugas akhir akan dinilai oleh pembimbing. Jika dinilai baik, mahasiswa berhak mengerjakan tugas akhir tersebut. Ujian pertama setelah fase pengerjaan tugas akhir adalah ujian kelayakan. Ujian kelayakan merupakan proses menilai kelayakan naskah tugas akhir yang dihadiri oleh dosen pembimbing dan penguji. Setelah lulus ujian kelayakan, mahasiswa berhak melanjutkan ke ujian akhir yang dapat dilaksanakan paling cepat 3 hari setelah ujian kelayakan. Ujian akhir merupakan proses presentasi dan pengujian tugas akhir yang dihadiri oleh dosen penguji dan pembimbing.



Gambar 2.4 Prosedur Tugas Akhir Jurusan Teknik Sistem Perkapalan – FTK

Prosedur tugas akhir Jurusan Teknik Sistem Perkapalan ditunjukkan pada Gambar 2.4. Mahasiswa memilih dosen pembimbing di awal proses. Bersama dosen pembimbing, mahasiswa diwajibkan membuat proposal pengajuan judul tugas akhir. Proposal tersebut akan dipresentasikan dalam sebuah pertemuan yang dihadiri oleh dosen pembimbing dan penguji. Kelulusan dalam presentasi pertama tersebut merupakan syarat mahasiswa untuk mengerjakan tugas akhir. Setelah cukup lama mengerjakan tugas akhir, mahasiswa harus melaksanakan presentasi progres yang dihadiri oleh dosen pembimbing dan penguji. Setelah

itu, mahasiswa harus menghadiri presentasi hasil yang merupakan ujian akhir bagi tugas akhir mahasiswa.



Gambar 2.5 Prosedur Tugas Akhir Jurusan Teknik Geomatika - FTSP

Prosedur tugas akhir Jurusan Teknik Geomatika ditunjukkan pada Gambar 2.5. Mahasiswa memilih dua orang dosen pembimbing di awal proses. Setelah mendapatkan dosen pembimbing, mahasiswa mengerjakan proposal untuk kemudian diuji dalam ujian proposal. Apabila lulus, mahasiswa berhak mengerjakan tugas akhirnya. Setelah tugas akhir selesai dikerjakan, mahasiswa harus mempresentasikan tugas akhirnya dalam seminar tugas akhir yang dihadiri oleh dosen pembimbing dan minimal sepuluh mahasiswa peserta seminar. Dari seminar tersebut, mahasiswa mendapatkan saran perbaikan yang harus diterapkan. Setelah memperbaiki tugas akhirnya, mahasiswa harus menghadiri ujian tugas akhir di mana karya mahasiswa diuji oleh dosen penguji. Hasil ujian tugas akhir tersebut menentukan kelulusan tugas akhir mahasiswa.

Dari kelima jurusan tersebut, hanya jurusan Teknik Informatika dan Teknik Mesin yang telah memanfaatkan teknologi informasi untuk mengelola proses pengerjaan tugas akhir mahasiswa. Kedua jurusan tersebut telah menggunakan perangkat lunak berbasis web berupa sistem informasi yang dikelola oleh masing-masing jurusan. Perangkat lunak tersebut digunakan oleh

mahasiswa, dosen, koordinator tugas akhir, dan petugas jurusan dalam rangka melaksanakan prosedur tugas akhir.

Sistem informasi tersebut telah dirancang agar sesuai dengan prosedur tugas akhir jurusan masing-masing. Sistem informasi tugas akhir milik Jurusan Teknik Informatika bersifat publik, yaitu dapat diakses dari jaringan manapun. Sistem tersebut dikembangkan dan dirawat oleh pihak jurusan. Sedangkan, sistem informasi tugas akhir milik Teknik Mesin masih bersifat lokal, yaitu hanya dapat diakses dari dalam jaringan ITS. Sistem tersebut dikembangkan dan dirawat oleh LPTSI ITS.

2.2 Organisasi dan Tata Kerja ITS

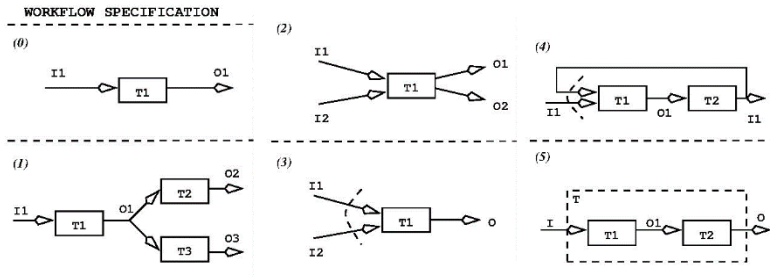
Organisasi dan Tata Kerja (OTK) ITS diatur dalam Permendikbud RI Nomor 86 tahun 2013. Pasal 59 peraturan perundang-undangan tersebut menyebutkan bahwa ITS terdiri dari enam fakultas, yaitu Fakultas Matematika dan Ilmu Pengetahuan Alam, Fakultas Teknologi Industri, Fakultas Teknik Sipil dan Perencanaan, Fakultas Teknologi Kelautan, Fakultas Teknologi Informasi, dan Fakultas Desain dan Industri Kreatif. Setiap jurusan di fakultas-fakultas tersebutlah yang menjadi subjek dari pembangunan sistem informasi administrasi tugas akhir ITS.

OTK ITS tersebut juga menyebutkan pelaksanaan penelitian dan pengembangan teknologi dan sistem informasi di ITS merupakan tugas dari Lembaga Pengembangan Teknologi Sistem Informasi (LPTSI) ITS. Oleh karena itu, kerja sama dengan LPTSI ITS akan dijalin dalam pembuatan tugas akhir ini [2].

2.3 Konsep Aliran Kerja

Aliran kerja pada dasarnya adalah tentang organisasi kerja. Aliran kerja adalah serangkaian kegiatan yang mengkoordinasikan orang dan / atau perangkat lunak. Aliran kerja dapat menjadi model yang menjadi jembatan komunikasi antara manusia dan proses otomatis. Hal tersebutlah yang menjadi nilai tambah aliran kerja. Aliran kerja bersifat fraktal. Ini berarti sebuah aliran kerja dapat terdiri dari aliran kerja lain. Model aliran kerja mendorong penggunaan kembali dan kelincahan sistem (*agility*), yang mengarah ke proses bisnis yang lebih fleksibel [3].

Spesifikasi aliran kerja terdiri dari masukan, proses, dan keluaran [4]. Ketiga elemen-elemen tersebut menyusun aliran kerja ke dalam beberapa bentuk atomik sebagaimana Gambar 2.6 berikut ini.



Gambar 2.6 Spesifikasi Aliran Kerja

2.4 Migrasi Sistem

Penggunaan secara luas dari teknologi komputer dalam beberapa dekade terakhir telah menghasilkan berbagai sistem yang besar dan kompleks. Sistem tersebut beberapa di antaranya telah berevolusi menjadi sistem yang menentang modifikasi dan evolusi lebih lanjut. Sistem seperti itu biasa disebut sebagai sistem warisan.

Migrasi sistem warisan diwujudkan dengan mengimplementasikan sebuah sistem target yang memiliki fungsionalitas dan data yang sama dengan sistem warisan, namun dapat dirawat dengan mudah dan dapat beradaptasi terhadap kebutuhan bisnis di masa depan. Strategi dan kerangka kerja seperti itu dibutuhkan untuk mendukung metode migrasi sistem yang ada [5].

2.5 Kerangka Kerja Pemrograman Laravel

Laravel adalah kerangka kerja pemrograman untuk aplikasi berbasis web. Laravel memiliki sintaksis yang ekspresif dan elegan. Kerangka kerja ini berusaha untuk menyinkronkan hal-hal yang menyusahkan dalam pengembangan dengan mempermudah pekerjaan-pekerjaan umum yang sering digunakan pada mayoritas aplikasi web. Hal-hal umum tersebut di antaranya adalah autentikasi, *routing*, pengaturan sesi, dan *caching* [6].

Laravel bertujuan untuk membuat proses pembangunan yang menyenangkan bagi pengembang tanpa mengorbankan fungsionalitas aplikasi. Pengembang yang senang menghasilkan kode program terbaik. Untuk tujuan ini, Laravel telah berusaha untuk menggabungkan yang terbaik dari apa yang telah ada dalam kerangka web lain, termasuk kerangka yang diimplementasikan dalam bahasa lain, seperti Ruby on Rails, ASP.NET MVC, dan Sinatra.

Laravel menyediakan alat-alat yang kuat yang diperlukan untuk aplikasi besar, aplikasi yang kuat. Sebuah inversi kontrol kontainer, sistem migrasi ekspresif, dan terintegrasi dukungan unit testing memberikan alat yang dibutuhkan pengembang untuk membangun aplikasi.

BAB 3

ANALISIS DAN PERANCANGAN SISTEM

Bab ini membahas tahap analisis permasalahan dan perancangan dari sistem yang akan dibangun yaitu SIMTA ITS. Analisis permasalahan membahas permasalahan yang diangkat dalam pengerjaan tugas akhir. Analisis kebutuhan mencantumkan kebutuhan-kebutuhan yang diperlukan perangkat lunak. Selanjutnya, yang dibahas adalah mengenai perancangan sistem yang dibuat. Pendekatan yang dibuat dalam perancangan ini adalah pendekatan berorientasi objek. Perancangan direpresentasikan dengan diagram UML (*Unified Modelling Language*).

3.1 Analisis

Tahap analisis dibagi menjadi beberapa bagian antara lain cakupan permasalahan, deskripsi umum sistem, kasus penggunaan sistem, dan kebutuhan perangkat lunak.

3.1.1 Analisis Permasalahan

Untuk mengetahui permasalahan dan kebutuhan pengguna sistem yang akan dibangun ini, diskusi bersama dan wawancara dilakukan dengan beberapa pihak terkait dari beberapa jurusan. Kegiatan diskusi dan wawancara tersebut telah dilakukan dengan bekerjasama dengan Pusat Pengembangan LPTSI selaku pihak yang akan melanjutkan perawatan sistem di masa yang akan datang. Pihak jurusan yang diundang dalam kegiatan tersebut di antaranya menjabat sebagai koordinator tugas akhir, ketua laboratorium, dan administrator SIM tugas akhir jurusan.

Berdasarkan diskusi dan wawancara tersebut, terdapat beberapa hal yang menjadi perhatian bersama. Beberapa di antaranya adalah sebagai berikut.

1. Tidak semua jurusan menggunakan sistem informasi dalam mengelola administrasi tugas akhir.
2. Koordinator tugas akhir di beberapa jurusan mengalami kesulitan dalam mengelola jadwal sidang.
3. Progres terkini mahasiswa sulit dipantau karena pelaporan progres masih dilakukan dengan pencatatan manual.

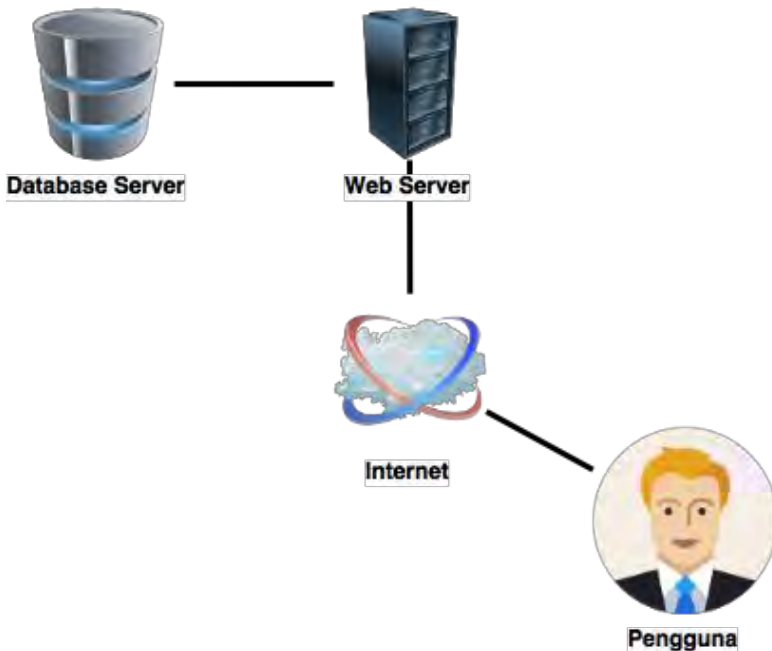
4. Prosedur tugas akhir di setiap jurusan berbeda antara satu dengan lainnya.
5. Sistem informasi yang dimiliki beberapa jurusan tertentu belum menghasilkan laporan yang baik.

Kelima poin tersebut merupakan permasalahan yang akan diselesaikan dalam konstruksi SIMTA ITS. Pembuatan sistem informasi yang bersifat generik adalah solusi agar terciptanya sistem yang terpadu. Dalam sistem tersebut fitur pengelolaan jadwal sidang dibutuhkan untuk mengelola baik sidang proposal maupun sidang akhir. Mahasiswa juga akan dapat memasukkan progress terkininya ke dalam sistem. Sistem akan dirancang agar dapat mengakomodasi beberapa perbedaan prosedur tugas akhir yang terdapat di beberapa jurusan. Kemudian, beberapa laman laporan akan dibuat guna memenuhi kebutuhan informasi bagi pihak jurusan.

3.1.2 Deskripsi Umum Sistem

Sistem informasi administrasi tugas akhir yang bersifat generik merupakan sistem informasi yang merangkum berbagai proses bisnis, dari berbagai jurusan di ITS, menjadi sebuah sistem tunggal dengan struktur data atau basis data yang sama. Setiap jurusan dapat menjalankan proses bisnisnya masing-masing di dalam sistem generik tersebut tanpa mempengaruhi proses bisnis yang dijalankan oleh jurusan lainnya.

Sistem diharapkan dapat memenuhi kebutuhan pengguna jurusan dalam menjalankan kegiatan terkait administrasi tugas akhir. Dengan sistem yang bersifat generik dan dibangun dengan fondasi kerangka kerja pemrograman yang baik, proses perawatan dan perkembangan sistem di masa depan dapat dilakukan dengan lebih mudah dan cepat.



Gambar 3.1 Gambaran Umum Perangkat Lunak

Perangkat lunak yang dibangun adalah berbasis web. Oleh karena itu dibutuhkan *web server* dan *database server* untuk memenuhi kebutuhan sistem. Gambar 3.1 menunjukkan arsitektur sistem.

Web server dan *database server* berada dalam dua fisik komputer *server* yang berbeda. Hal tersebut ditujukan agar sistem memiliki kehandalan yang baik dalam melayani permintaan akses web dari pengguna.

3.1.3 Spesifikasi Kebutuhan Perangkat Lunak

Pada sistem yang akan dibuat ini, dibutuhkan beberapa fungsi yang dapat membantu proses bisnis dalam sistem. Fungsi-fungsi tersebut antara lain adalah sebagai berikut.

1. Pengelolaan data pengguna.

2. Pengelolaan data pembimbing.
3. Pengelolaan data tugas akhir.
4. Pengelolaan progres bimbingan dan buku tugas akhir.
5. Pengelolaan jadwal sidang.
6. Pengelolaan nilai tugas akhir.
7. Pengelolaan laporan-laporan tugas akhir.

3.1.4 Aktor

Aktor mendefinisikan entitas-entitas yang terlibat dan berinteraksi langsung dengan sistem. Entitas ini bisa berupa manusia maupun sistem atau perangkat lunak yang lain. Aktor yang terdapat pada sistem ini hanya memiliki sebuah peran yaitu sebagai pengguna.

Identifikasi aktor dilakukan berdasarkan hasil wawancara dengan pihak jurusan. Berdasarkan hasil wawancara yang telah dilakukan aktor-aktor yang teridentifikasi yaitu mahasiswa, dosen, koordinator tugas akhir, dan administrator. Mahasiswa dan dosen merupakan aktor-aktor utama yang berkepentingan dalam menjalankan prosedur tugas akhir. Koordinator tugas akhir berperan sebagai manajer dalam mengelola jalannya kegiatan administrasi tugas akhir. Sedangkan, administrator berperan sebagai pembantu koordinator tugas akhir dalam menyelesaikan urusan administrasi mengenai tugas akhir.

Selain keempat aktor tersebut, terdapat aktor yang bertugas mengelola aliran kerja sistem. Aktor ini adalah administrator pusat, yang memiliki wewenang mengatur data-data master yang berkaitan dengan aliran kerja sistem.

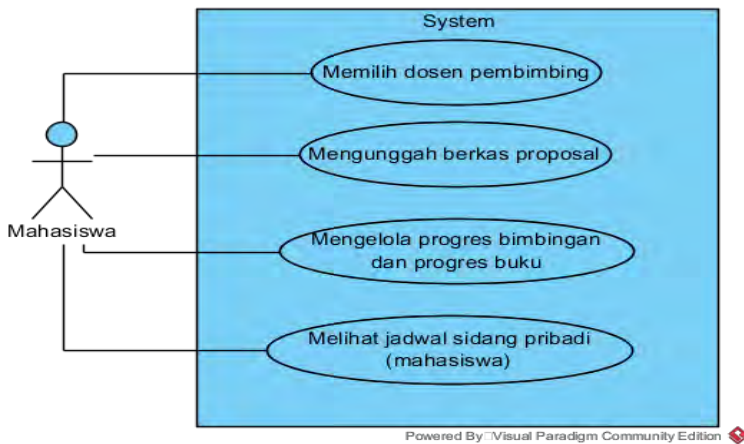
3.1.5 Kasus Penggunaan

Kasus penggunaan sistem ini dianalisis berdasarkan hasil wawancara dengan pihak jurusan. Prosedur tugas akhir dari beberapa jurusan yang telah diwawancarai dianalisis untuk menghasilkan sebuah kasus penggunaan yang generik. Tabel 3.1 berikut ini merupakan rincian kasus-kasus penggunaan generik tersebut. Masing-masing kasus penggunaan akan dijelaskan pada

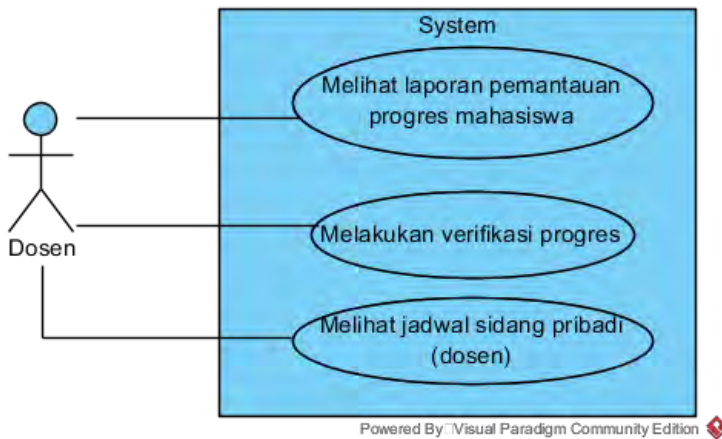
subbab tersendiri. Diagram kasus penggunaan untuk setiap aktor ditunjukkan oleh Gambar 3.2 hingga Gambar 3.6.

Tabel 3.1 Daftar Kasus Penggunaan

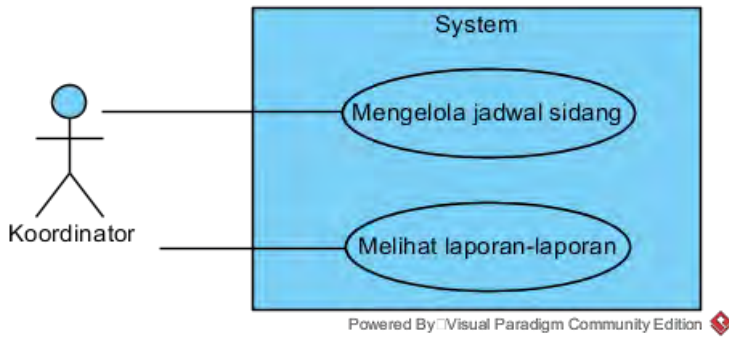
Aktor	Kode Kasus Penggunaan	Kasus Penggunaan
Mahasiswa	FR-01	Memilih dosen pembimbing
	FR-02	Mengunggah berkas proposal
	FR-03	Mengelola progres bimbingan dan progres buku
	FR-04	Melihat jadwal sidang pribadi (mahasiswa)
Dosen	FR-05	Melihat laporan pemantauan progres mahasiswa
	FR-06	Melakukan verifikasi progres
	FR-07	Melihat jadwal sidang pribadi (dosen)
Koordinator	FR-08	Mengelola jadwal sidang
	FR-09	Melihat laporan-laporan
Administrator	FR-10	Mengelola data mahasiswa
	FR-11	Mengelola nilai tugas akhir
Administrator Pusat	FR-12	Mengelola jurusan
	FR-13	Mengelola aliran kerja
	FR-14	Mengelola hak akses
	FR-15	Mengelola parameter
	FR-16	Mengelola menu
	FR-17	Mengelola peran



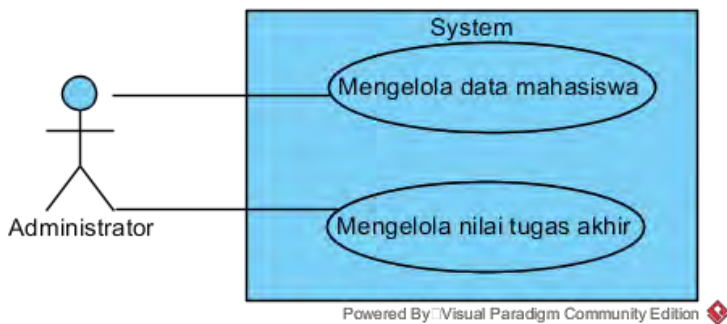
Gambar 3.2 Kasus Penggunaan Aktor Mahasiswa



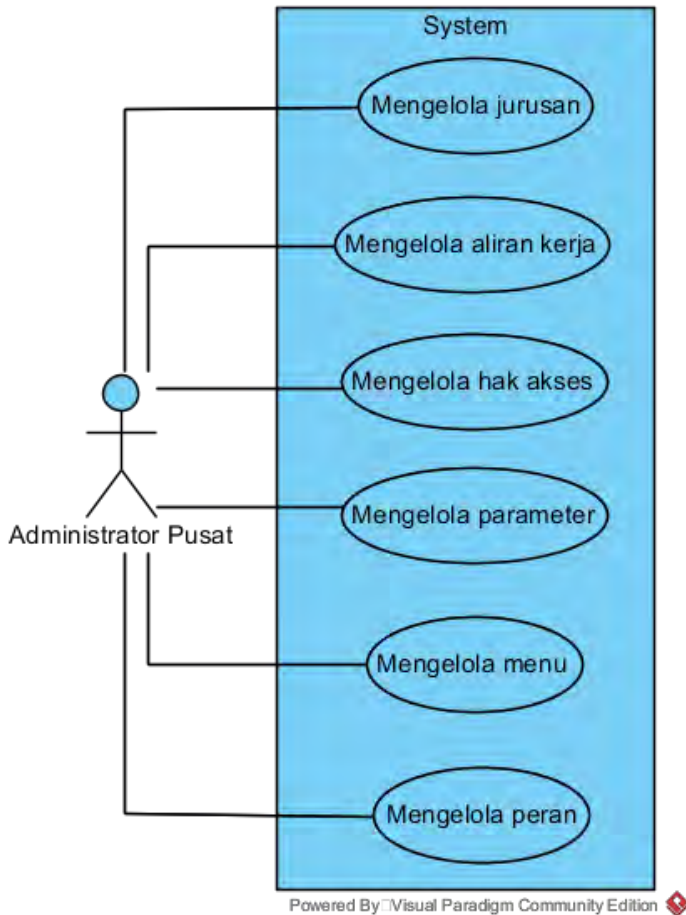
Gambar 3.3 Kasus Penggunaan Aktor Dosen



Gambar 3.4 Kasus Penggunaan Aktor Koordinator



Gambar 3.5 Kasus Penggunaan Aktor Administrator



Gambar 3.6 Kasus Penggunaan Aktor Administrator Pusat

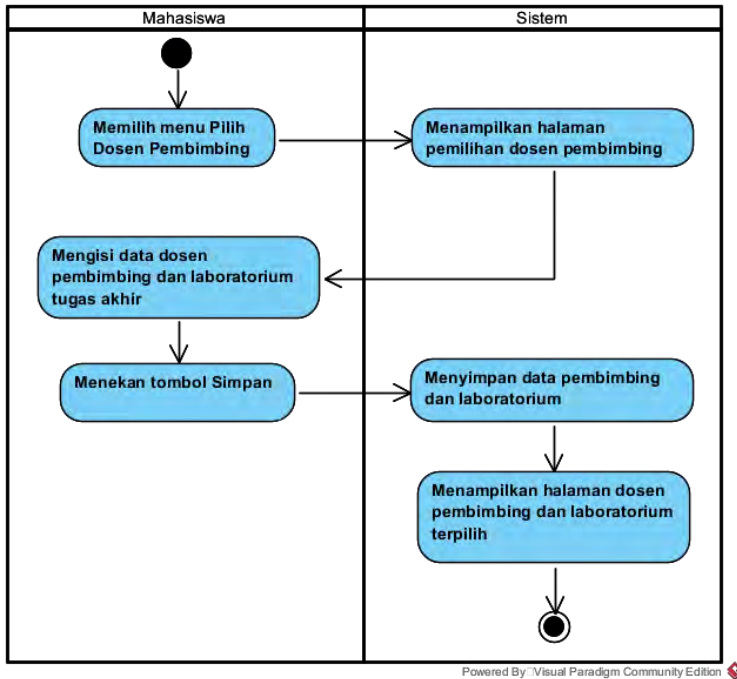
3.1.5.1 Memilih Dosen Pembimbing

Mahasiswa dapat menentukan dosen pembimbing untuk tugas akhirnya berdasarkan daftar dosen yang tersedia di jurusannya. Saat memilih dosen pembimbing, mahasiswa juga

harus menentukan laboratorium tempat pengerjaan tugas akhir. Spesifikasi kasus penggunaan memilih dosen pembimbing dapat dilihat pada Tabel 3.2. Selain itu, Gambar 3.7 menunjukkan diagram aktivitas yang mendefinisikan aktivitas pengguna dan respon yang diberikan sistem.

Tabel 3.2 Spesifikasi Kasus Penggunaan Memilih Dosen Pembimbing

Nama	Memilih Dosen Pembimbing
Kode	FR-01
Deskripsi	Memilih dosen sebagai pembimbing tugas akhir.
Tipe	Fungsional
Pemicu	Mahasiswa memilih menu untuk memilih dosen pembimbing
Aktor	Mahasiswa
Kondisi Awal	Mahasiswa telah login
Aliran: - Kejadian Normal	<ol style="list-style-type: none"> 1. Mahasiswa melakukan aksi untuk memilih dosen pembimbing. 2. Sistem menampilkan halaman pemilihan dosen pembimbing. 3. Mahasiswa mengisi data dosen pembimbing dan laboratorium tugas akhir. 4. Mahasiswa melakukan aksi untuk menyimpan data. 5. Sistem menyimpan data.
- Kejadian Alternatif	-
Kondisi Akhir	Sistem menampilkan data yang dosen pembimbing dan laboratorium tugas akhir.



Gambar 3.7 Diagram Aktivitas Memilih Dosen Pembimbing

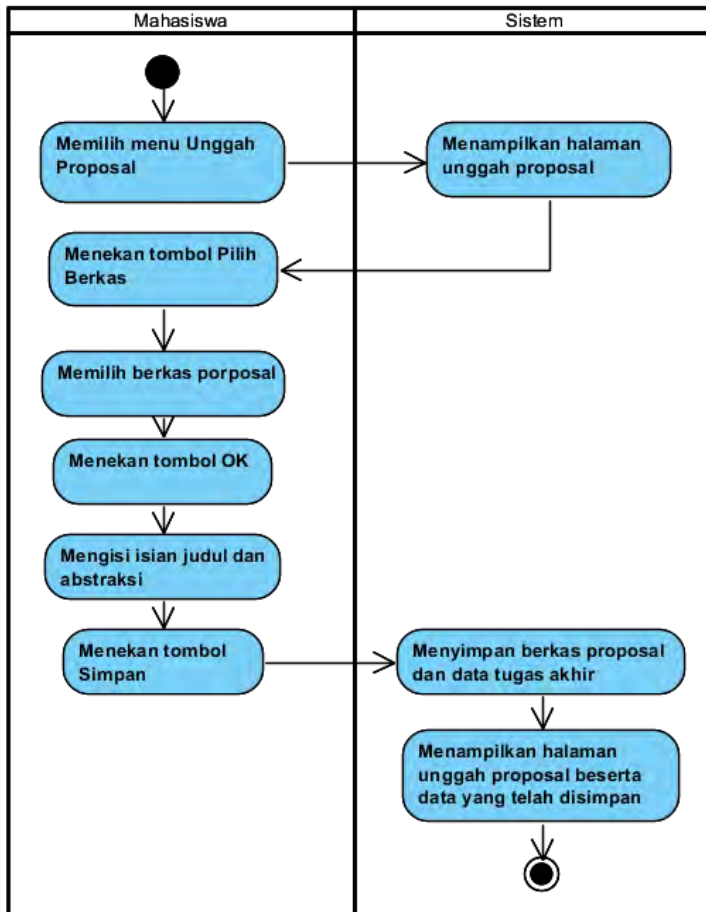
3.1.5.2 Mengunggah Berkas Proposal

Mahasiswa yang telah memilih dosen pembimbing berhak melakukan pengunggahan berkas proposal. Saat mengunggah proposal, mahasiswa diwajibkan mengisi judul beserta abstraksi tugas akhir. Spesifikasi penggunaan mengunggah berkas proposal dapat dilihat pada Tabel 3.3. Selain itu, Gambar 3.8 menunjukkan diagram aktivitas yang mendefinisikan aktivitas pengguna dan respon yang diberikan sistem.

Tabel 3.3 Spesifikasi Kasus Penggunaan Mengunggah Berkas Proposal

Nama	Mengunggah berkas proposal.
Kode	FR-02

Deskripsi	Mahasiswa mengunggah berkas proposal serta mengisi data rincian tugas akhir, yaitu judul dan abstraksi tugas akhir.
Tipe	Fungsional
Pemicu	Mahasiswa memilih untuk mengunggah berkas proposal.
Aktor	Mahasiswa
Kondisi Awal	Mahasiswa telah login
Aliran: - Kejadian Normal	1. Mahasiswa melakukan aksi untuk mengunggah berkas proposal. 2. Sistem menampilkan halaman pengunggahan berkas proposal. 3. Mahasiswa memilih berkas proposal untuk diunggah serta mengisi isian judul dan abstraksi tugas akhir. 4. Mahasiswa melakukan aksi untuk menyimpan data. 5. Sistem menyimpan proposal yang diunggah.
- Kejadian Alternatif	-
Kondisi Akhir	Sistem menampilkan proposal yang telah diunggah.



Powered By Visual Paradigm Community Edition

Gambar 3.8 Diagram Aktivitas Mengunggah Berkas Proposal

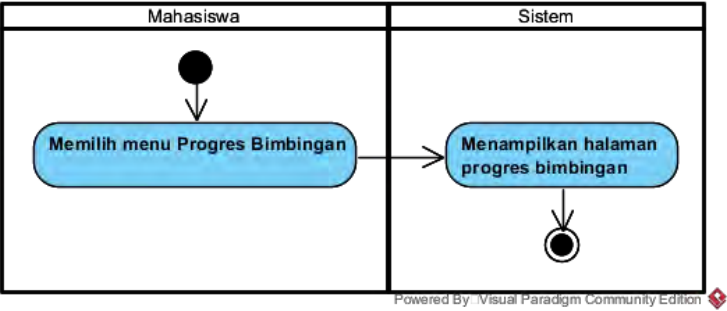
3.1.5.3 Mengelola Progres Bimbingan Dan Progres Buku

Mahasiswa melaporkan progres pengerjaan tugas akhir dalam dua entitas, yaitu progres bimbingan dan progres buku. Progres bimbingan merupakan laporan kemajuan yang bersifat

deskriptif mengenai status tugas akhir. Sedangkan, progres buku merupakan laporan seberapa jauh mahasiswa telah menyelesaikan buku tugas akhir. Kasus penggunaan ini terjabarkan dalam beberapa subkasus penggunaan. Spesifikasi-spesifikasi penggunaan mengelola progres bimbingan dan progres buku dapat dilihat pada Tabel 3.4 hingga Tabel 3.10. Selain itu, Gambar 3.9 hingga Gambar 3.15 menunjukkan diagram aktivitas yang mendefinisikan aktivitas pengguna dan respon yang diberikan sistem.

Tabel 3.4 Spesifikasi Kasus Penggunaan Melihat Progres Bimbingan Dan Progres Buku

Nama	Melihat progres bimbingan dan progres buku
Kode	FR-03.1
Deskripsi	Mahasiswa melihat daftar progres bimbingan dan progres buku dalam sebuah halaman.
Tipe	Fungsional
Pemicu	Mahasiswa memilih untuk melihat daftar progres bimbingan dan progres buku.
Aktor	Mahasiswa
Kondisi Awal	Mahasiswa telah login
Aliran: - Kejadian Normal	<ol style="list-style-type: none"> 1. Mahasiswa melakukan aksi untuk melihat daftar progres bimbingan dan progres buku. 2. Sistem menampilkan halaman daftar progres bimbingan dan progres buku mahasiswa.
- Kejadian Alternatif	-
Kondisi Akhir	Sistem menampilkan halaman daftar progres bimbingan dan progres buku mahasiswa.

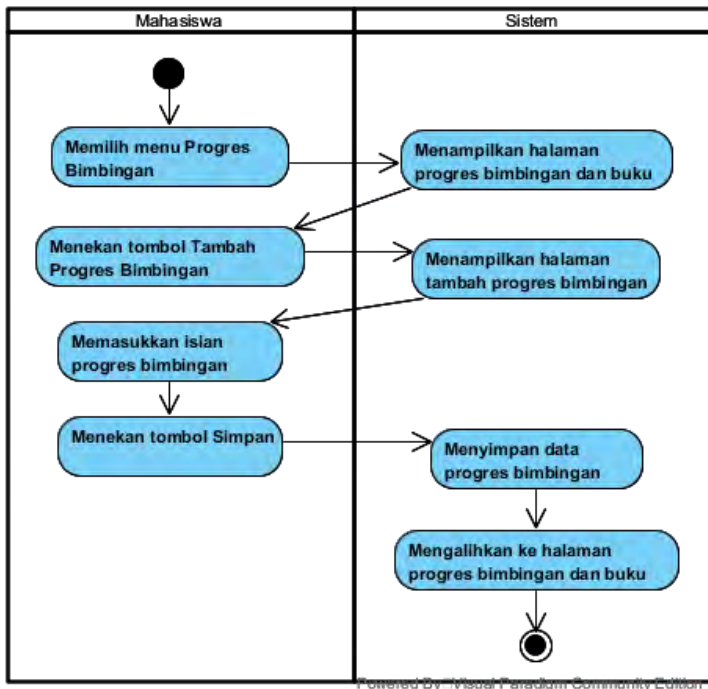


Gambar 3.9 Diagram Aktivitas Melihat Progres Bimbingan dan Progres Buku

Tabel 3.5 Spesifikasi Kasus Penggunaan Menambahkan Progres Bimbingan

Nama	Menambahkan progres bimbingan
Kode	FR-03.2
Deskripsi	Mahasiswa menambahkan progres bimbingan
Tipe	Fungsional
Pemicu	Mahasiswa memilih untuk menambahkan progres bimbingan.
Aktor	Mahasiswa
Kondisi Awal	Mahasiswa telah login
Aliran: - Kejadian Normal	<ol style="list-style-type: none">1. Mahasiswa melakukan aksi untuk membuka halaman progres bimbingan.2. Mahasiswa menekan tombol progres bimbingan.3. Sistem menampilkan halaman isian progres bimbingan.4. Mahasiswa memasukkan data progres bimbingan yang meliputi tanggal dan aktivitas bimbingan.5. Mahasiswa melakukan aksi untuk menyimpan data.

	6. Sistem menyimpan data dan mengarahkan mahasiswa ke halaman daftar progres bimbingan dan progres buku.
- Kejadian Alternatif	-
Kondisi Akhir	Data progres bimbingan mahasiswa telah disimpan.

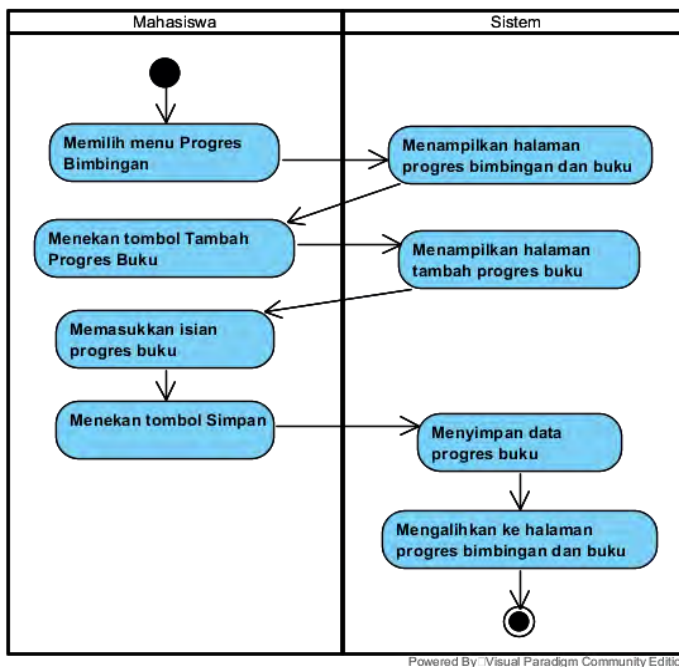


Gambar 3.10 Diagram Aktivitas Menambahkan Progres Bimbingan

Tabel 3.6 Spesifikasi Kasus Penggunaan Menambahkan Progres Buku

Nama	Menambahkan progres buku
Kode	FR-03.3

Deskripsi	Mahasiswa menambahkan progres buku
Tipe	Fungsional
Pemicu	Mahasiswa memilih untuk menambahkan progres buku.
Aktor	Mahasiswa
Kondisi Awal	Mahasiswa telah login
Aliran: - Kejadian Normal	<ol style="list-style-type: none"> 1. Mahasiswa melakukan aksi untuk membuka halaman progres buku. 2. Mahasiswa menekan tombol progres buku. 3. Sistem menampilkan halaman isian progres buku. 4. Mahasiswa memasukkan data progres bimbingan yang meliputi tanggal, bab, pencapaian, berkas buku, dan penjelasan. 5. Mahasiswa melakukan aksi untuk menyimpan data. 6. Sistem menyimpan data dan mengarahkan mahasiswa ke halaman daftar progres bimbingan dan progres buku.
- Kejadian Alternatif	-
Kondisi Akhir	Data progres buku mahasiswa telah disimpan.

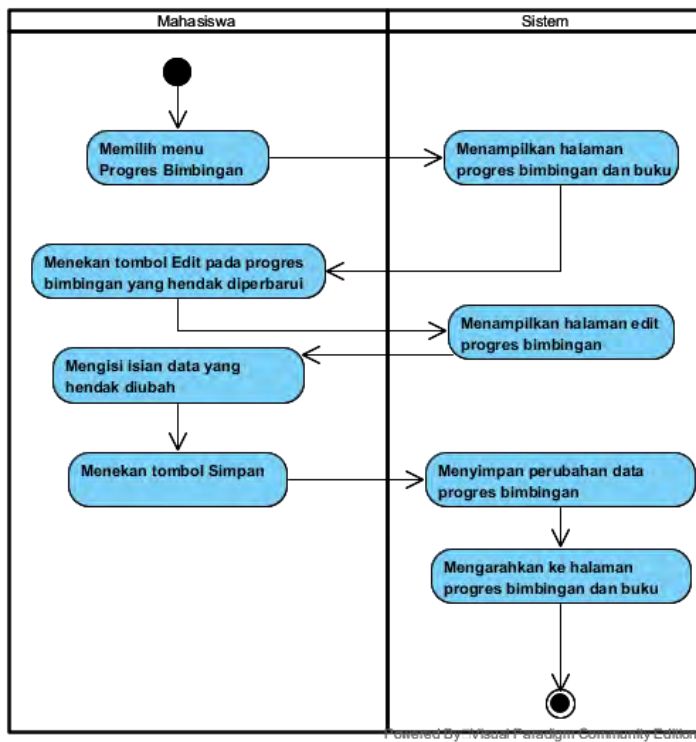


Gambar 3.11 Diagram Aktivitas Menambahkan Progres Buku

Tabel 3.7 Spesifikasi Kasus Penggunaan Memperbarui Progres Bimbingan

Nama	Memperbarui progres bimbingan
Kode	FR-03.4
Deskripsi	Mahasiswa memperbarui data progres bimbingan yang telah ditambahkan sebelumnya
Tipe	Fungsional
Pemicu	Mahasiswa memilih untuk memperbarui data progres bimbingan
Aktor	Mahasiswa
Kondisi Awal	Mahasiswa telah login

Aliran: - Kejadian Normal	1. Mahasiswa melakukan aksi untuk membuka halaman progres bimbingan. 2. Mahasiswa menekan tombol <i>edit</i> pada progres yang hendak diperbarui. 3. Sistem menampilkan halaman untuk memperbarui data progres bimbingan. 4. Mahasiswa melakukan perubahan pada isian yang hendak diubah. 5. Mahasiswa melakukan aksi untuk menyimpan data. 6. Sistem menyimpan data dan mengarahkan mahasiswa ke halaman daftar progres bimbingan dan progres buku.
- Kejadian Alternatif	-
Kondisi Akhir	Data progres bimbingan mahasiswa telah berhasil diubah.

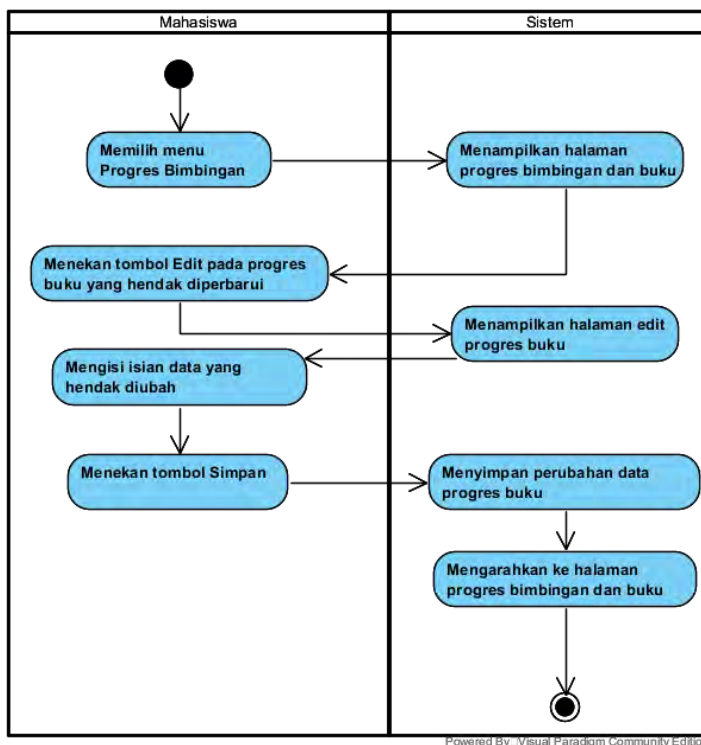


Gambar 3.12 Diagram Aktivitas Memperbarui Progres Bimbingan

Tabel 3.8 Spesifikasi Kasus Penggunaan Memperbarui Progres Buku

Nama	Memperbarui progres buku
Kode	FR-03.5
Deskripsi	Mahasiswa memperbarui data progres buku yang telah ditambahkan sebelumnya
Tipe	Fungsional
Pemicu	Mahasiswa memilih untuk memperbarui data progres buku
Aktor	Mahasiswa

Kondisi Awal	Mahasiswa telah login
Aliran: - Kejadian Normal	<ol style="list-style-type: none"> 1. Mahasiswa melakukan aksi untuk membuka halaman progres buku. 2. Mahasiswa menekan tombol <i>edit</i> pada progres buku yang hendak diperbarui. 3. Sistem menampilkan halaman untuk memperbarui data progres buku. 4. Mahasiswa melakukan perubahan pada isian yang hendak diubah. 5. Mahasiswa melakukan aksi untuk menyimpan data. 6. Sistem menyimpan data dan mengarahkan mahasiswa ke halaman daftar progres bimbingan dan progres buku.
- Kejadian Alternatif	-
Kondisi Akhir	Data progres buku mahasiswa telah berhasil diubah.

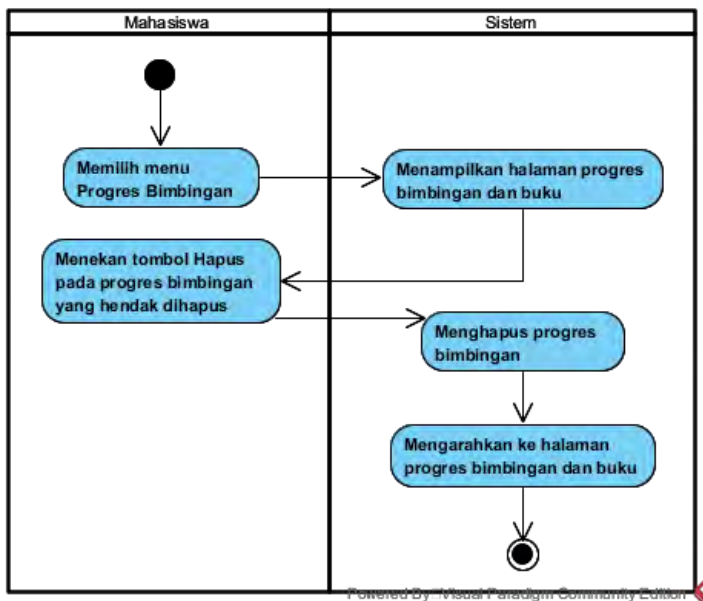


Gambar 3.13 Diagram Aktivitas Memperbarui Progres Buku

Tabel 3.9 Spesifikasi Kasus Penggunaan Menghapus Progres Bimbingan

Nama	Menghapus progres bimbingan
Kode	FR-03.6
Deskripsi	Mahasiswa menghapus data progres bimbingan yang telah ditambahkan sebelumnya
Tipe	Fungsional
Pemicu	Mahasiswa memilih untuk menghapus data progres bimbingan
Aktor	Mahasiswa

Kondisi Awal	Mahasiswa telah login
Aliran: - Kejadian Normal	1. Mahasiswa melakukan aksi untuk membuka halaman progres bimbingan. 2. Mahasiswa menekan tombol <i>delete</i> pada progres yang hendak dihapus. 3. Sistem menampilkan konfirmasi penghapusan. 4. Mahasiswa melakukan aksi untuk mengonfirmasikan penghapusan. 5. Sistem menghapus data dan mengarahkan mahasiswa ke halaman daftar progres bimbingan dan progres buku.
- Kejadian Alternatif	-
Kondisi Akhir	Data progres bimbingan mahasiswa telah berhasil dihapus.

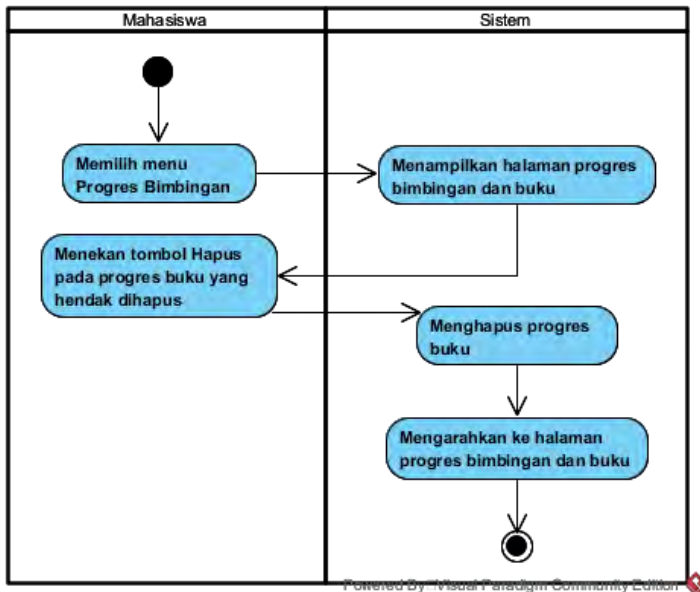


Gambar 3.14 Diagram Aktivitas Mengapus Progres Bimbingan

Tabel 3.10 Spesifikasi Kasus Penggunaan Menghapus Progres Buku

Nama	Menghapus progres buku
Kode	FR-03.7
Deskripsi	Mahasiswa menghapus data progres buku yang telah ditambahkan sebelumnya
Tipe	Fungsional
Pemicu	Mahasiswa memilih untuk menghapus data progres buku
Aktor	Mahasiswa
Kondisi Awal	Mahasiswa telah login
Aliran: - Kejadian Normal	<ol style="list-style-type: none"> 1. Mahasiswa melakukan aksi untuk membuka halaman progres buku. 2. Mahasiswa menekan tombol <i>delete</i> pada progres buku yang hendak dihapus.

	<ol style="list-style-type: none"> 3. Sistem menampilkan konfirmasi penghapusan. 4. Mahasiswa melakukan aksi untuk mengonfirmasikan penghapusan. 5. Sistem menghapus data dan mengarahkan mahasiswa ke halaman daftar progres bimbingan dan progres buku.
- Kejadian Alternatif	-
Kondisi Akhir	Data progres buku mahasiswa telah berhasil dihapus.



Gambar 3.15 Diagram Aktivitas Menghapus Progres Buku

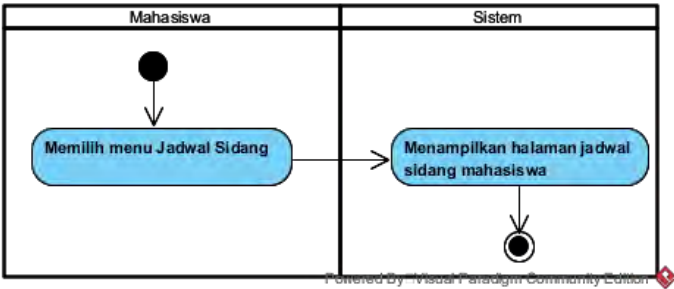
3.1.5.4 Melihat Jadwal Sidang Pribadi (Mahasiswa)

Mahasiswa dapat mengetahui kapan tanggal sidang proposal atau sidang akhir tugas akhir mereka diadakan. Informasi tersebut dapat dilihat halaman jadwal sidang pribadi. Spesifikasi

penggunaan melihat jadwal sidang pribadi mahasiswa dapat dilihat pada Tabel 3.11. Selain itu, Gambar 3.16 menunjukkan diagram aktivitas yang mendefinisikan aktivitas pengguna dan respon yang diberikan sistem.

Tabel 3.11 Spesifikasi Kasus Penggunaan Melihat Jadwal Sidang Pribadi (Mahasiswa)

Nama	Melihat jadwal sidang pribadi (mahasiswa)
Kode	FR-04
Deskripsi	Mahasiswa melihat jadwal sidang proposal dan sidang tugas akhir yang harus dihadiri
Tipe	Fungsional
Pemicu	Mahasiswa memilih untuk melihat jadwal sidang
Aktor	Mahasiswa
Kondisi Awal	Mahasiswa telah login
Aliran: - Kejadian Normal	1. Mahasiswa melakukan aksi untuk melihat jadwal sidang pribadi. 2. Sistem menampilkan halaman jadwal sidang mahasiswa yang bersangkutan.
- Kejadian Alternatif	-
Kondisi Akhir	Sistem menampilkan halaman jadwal sidang mahasiswa.



Gambar 3.16 Diagram Aktivitas Melihat Jadwal Sidang Pribadi (Mahasiswa)

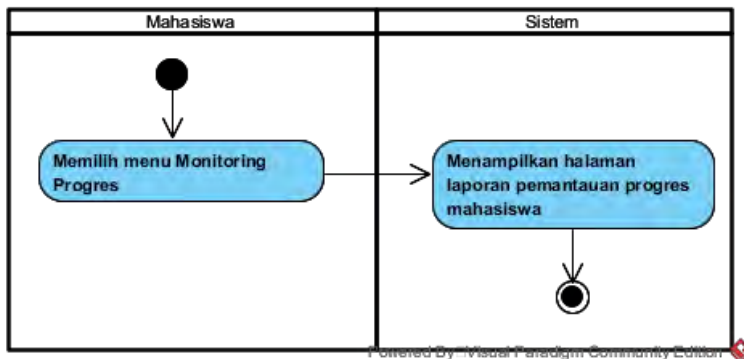
3.1.5.5 Melihat Laporan Pemantauan Progres Mahasiswa

Dosen dapat memantau progres pengerjaan tugas akhir mahasiswa bimbingannya melalui laporan pemantauan progres mahasiswa. Spesifikasi penggunaan melakukan verifikasi progres dapat dilihat pada Tabel 3.12. Selain itu, Gambar 3.17 menunjukkan diagram aktivitas yang mendefinisikan aktivitas pengguna dan respon yang diberikan sistem.

Tabel 3.12 Spesifikasi Kasus Penggunaan Melihat Laporan Pemantauan Progres Mahasiswa

Nama	Melihat laporan pemantauan progres mahasiswa
Kode	FR-05
Deskripsi	Dosen melihat laporan progres pengerjaan tugas akhir mahasiswa bimbingannya
Tipe	Fungsional
Pemicu	Dosen memilih untuk melihat laporan pemantauan progres mahasiswa
Aktor	Dosen
Kondisi Awal	Dosen telah login

Aliran: - Kejadian Normal	1. Dosen melakukan aksi untuk melihat laporan pemantauan progres mahasiswa. 2. Sistem menampilkan halaman laporan pemantauan progres mahasiswa.
- Kejadian Alternatif	-
Kondisi Akhir	Sistem menampilkan halaman laporan pemantauan progres mahasiswa.



Gambar 3.17 Diagram Aktivitas Melihat Laporan Pemantauan Progres Mahasiswa

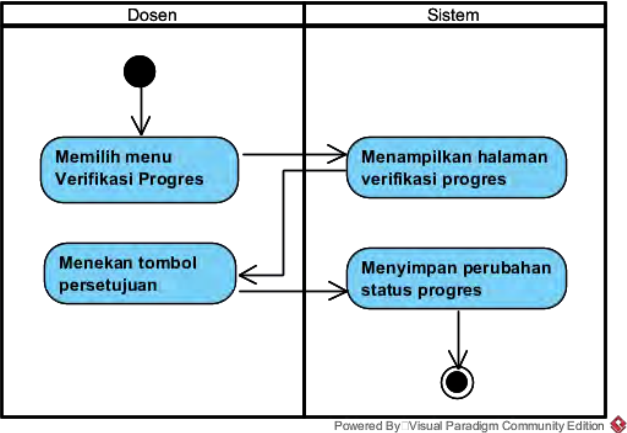
3.1.5.6 Melakukan Verifikasi Progres

Progres bimbingan dan progres buku yang dilaporkan oleh mahasiswa harus diverifikasi oleh dosen pembimbing mahasiswa. Verifikasi tersebut dapat dilakukan oleh dosen di halaman verifikasi progres. Spesifikasi penggunaan melakukan verifikasi progres dapat dilihat pada

Tabel 3.13. Selain itu, Gambar 3.18 menunjukkan diagram aktivitas yang mendefinisikan aktivitas pengguna dan respon yang diberikan sistem.

Tabel 3.13 Spesifikasi Kasus Penggunaan Melakukan Verifikasi Progres

Nama	Melakukan verifikasi progres
Kode	FR-06
Deskripsi	Dosen melakukan verifikasi progres mahasiswa bimbingannya
Tipe	Fungsional
Pemicu	Dosen memilih untuk melakukan verifikasi progres
Aktor	Mahasiswa
Kondisi Awal	Dosen telah login
Aliran: - Kejadian Normal	<ol style="list-style-type: none"> 1. Dosen melakukan aksi untuk melakukan verifikasi progres. 2. Sistem menampilkan halaman daftar verifikasi progres. 3. Dosen melakukan aksi untuk memberikan verifikasi dengan menekan tombol Verifikasi atau Tidak Valid 4. Sistem mengubah status verifikasi progres sesuai dengan verifikasi dari dosen.
- Kejadian Alternatif	-
Kondisi Akhir	Status verifikasi progres telah berhasil diubah sesuai dengan verifikasi dari dosen.



Gambar 3.18 Diagram Aktivitas Melakukan Verifikasi Progres

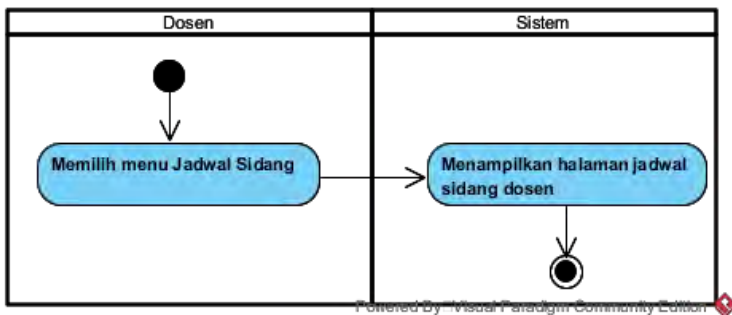
3.1.5.7 Melihat Jadwal Sidang Pribadi (Dosen)

Dosen dapat melihat jadwal sidang yang harus dihadiri, baik sidang proposal maupun sidang akhir. Untuk melihat jadwal sidang, dosen dapat membuka halaman jadwal sidang dosen. Spesifikasi penggunaan melihat jadwal sidang pribadi dosen dapat dilihat pada Tabel 3.14. Selain itu, Gambar 3.19 menunjukkan diagram aktivitas yang mendefinisikan aktivitas pengguna dan respon yang diberikan sistem.

Tabel 3.14 Spesifikasi Kasus Penggunaan Melihat Jadwal Sidang Pribadi (Dosen)

Nama	Melihat jadwal sidang pribadi (dosen)
Kode	FR-07
Deskripsi	Dosen melihat jadwal sidang proposal dan sidang tugas akhir yang harus dihadiri
Tipe	Fungsional
Pemicu	Dosen memilih untuk melihat jadwal sidang
Aktor	Dosen
Kondisi Awal	Dosen telah login

Aliran: - Kejadian Normal	1. Dosen melakukan aksi untuk melihat jadwal sidang pribadi. 2. Sistem menampilkan halaman jadwal sidang yang harus dihadiri oleh dosen yang bersangkutan.
- Kejadian Alternatif	-
Kondisi Akhir	Sistem menampilkan halaman jadwal sidang yang harus dihadiri oleh dosen yang bersangkutan.



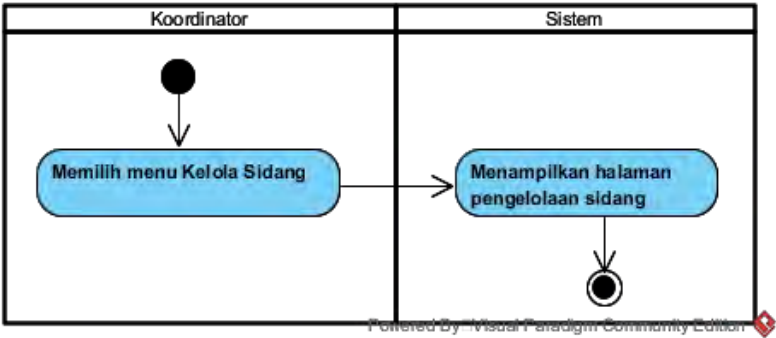
Gambar 3.19 Diagram Aktivitas Melihat Jadwal Sidang Pribadi (dosen)

3.1.5.8 Mengelola Jadwal Sidang

Jadwal sidang dikelola oleh koordinator tugas akhir. Untuk mengelola jadwal sidang, halaman yang dapat diakses adalah halaman kelola jadwal sidang. Dalam kasus penggunaan ini terdapat beberapa subkasus penggunaan yang menjelaskan rincian kasus penggunaan yang dapat diakses oleh koordinator. Spesifikasi-spesifikasi kasus penggunaan mengelola jadwal sidang dapat dilihat pada Tabel 3.15 hingga Tabel 3.18. Selain itu, Gambar 3.20 hingga Gambar 3.23 menunjukkan diagram aktivitas yang mendefinisikan aktivitas pengguna dan respon yang diberikan sistem.

Tabel 3.15 Spesifikasi Kasus Penggunaan Melihat Halaman Kelola Sidang

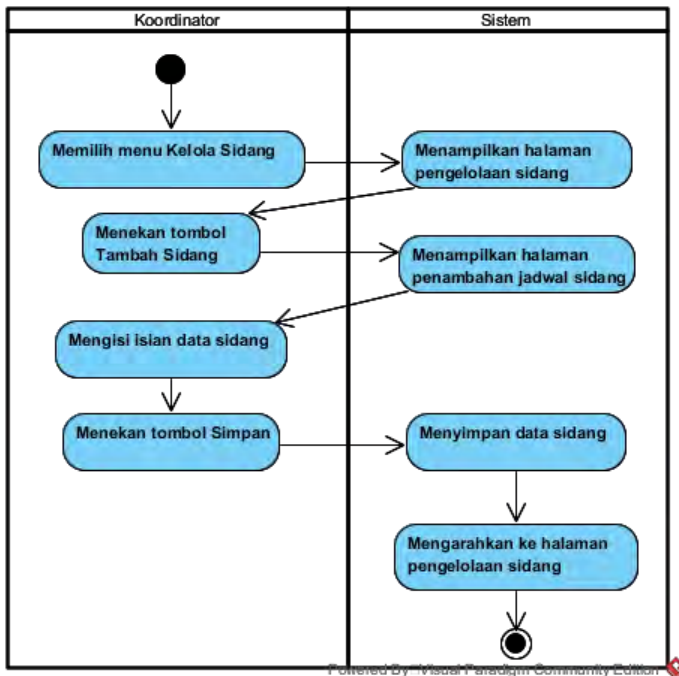
Nama	Melihat halaman kelola sidang
Kode	FR-08.1
Deskripsi	Koordinator membuka halaman kelola sidang yang berisi jadwal sidang beserta fitur-fitur pengelolaan seperti penambahan, pengubahan, dan penghapusan sidang.
Tipe	Fungsional
Pemicu	Koordinator memilih untuk membuka halaman kelola jadwal sidang.
Aktor	Koordinator
Kondisi Awal	Koordinator telah login
Aliran: - Kejadian Normal	1. Koordinator melakukan aksi untuk membuka halaman kelola jadwal sidang. 2. Sistem menampilkan halaman kelola jadwal sidang.
- Kejadian Alternatif	-
Kondisi Akhir	Sistem menampilkan halaman kelola jadwal sidang.



Gambar 3.20 Diagram Aktivitas Melihat Halaman Kelola Sidang

Tabel 3.16 Spesifikasi Kasus Penggunaan Menambahkan Jadwal Sidang

Nama	Menambahkan jadwal sidang
Kode	FR-08.2
Deskripsi	Koordinator menambahkan jadwal sidang
Tipe	Fungsional
Pemicu	Koordinator memilih untuk menambahkan jadwal sidang
Aktor	Koordinator
Kondisi Awal	Koordinator telah login
Aliran: - Kejadian Normal	<ol style="list-style-type: none"> 1. Koordinator melakukan aksi untuk membuka halaman kelola jadwal. 2. Koordinator menekan tombol tambah jadwal sidang. 3. Sistem menampilkan halaman isian penambahan jadwal sidang. 4. Koordinator mengisi isian jadwal sidang. 5. Koordinator melakukan aksi untuk menyimpan data jadwal sidang. 6. Sistem menyimpan data dan mengarahkan koordinator kembali ke halaman kelola jadwal sidang.
- Kejadian Alternatif	-
Kondisi Akhir	Data jadwal sidang telah disimpan.

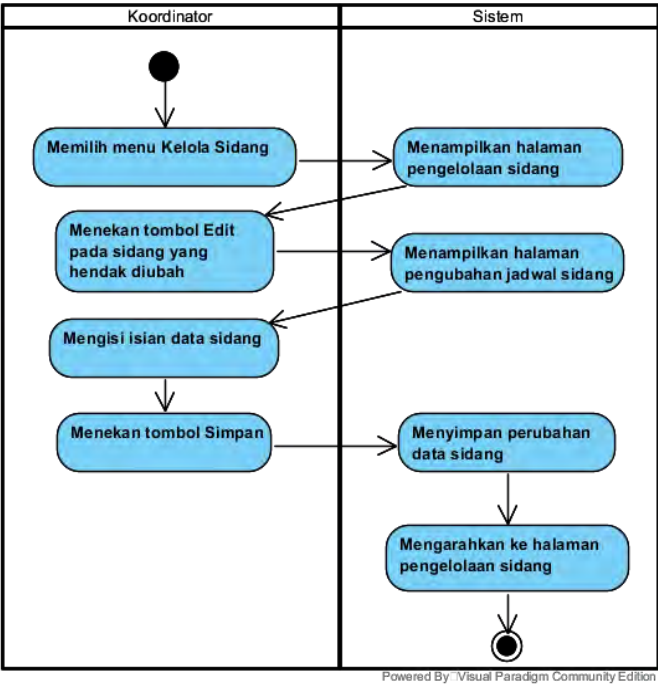


Gambar 3.21 Diagram Aktivitas Menambahkan Jadwal Sidang

Tabel 3.17 Spesifikasi Kasus Penggunaan Mengubah Data Jadwal Sidang

Nama	Mengubah data jadwal sidang
Kode	FR-08.3
Deskripsi	Koordinator mengubah jadwal sidang
Tipe	Fungsional
Pemicu	Koordinator memilih untuk mengubah jadwal sidang
Aktor	Koordinator
Kondisi Awal	Koordinator telah login

Aliran: - Kejadian Normal	<ol style="list-style-type: none"> 1. Koordinator melakukan aksi untuk membuka halaman kelola jadwal. 2. Koordinator menekan tombol <i>edit</i> jadwal sidang pada sidang yang akan diubah datanya. 3. Sistem menampilkan halaman perubahan jadwal sidang. 4. Koordinator mengubah isian jadwal sidang. 5. Koordinator melakukan aksi untuk menyimpan data jadwal sidang. 6. Sistem menyimpan data dan mengarahkan koordinator kembali ke halaman kelola jadwal sidang.
- Kejadian Alternatif	-
Kondisi Akhir	Data jadwal sidang telah berhasil diubah.



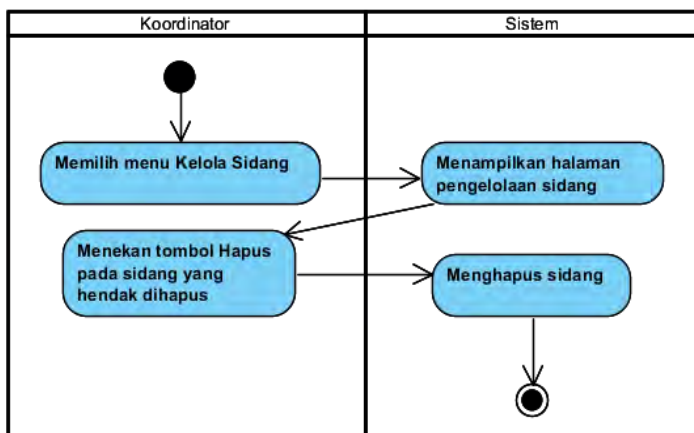
Powered By: Visual Paradigm Community Edition

Gambar 3.22 Diagram Aktivitas Mengubah Data Jadwal Sidang

Tabel 3.18 Spesifikasi Kasus Penggunaan Menghapus Jadwal Sidang

Nama	Menghapus jadwal sidang
Kode	FR-08.4
Deskripsi	Koordinator menghapus jadwal sidang
Tipe	Fungsional
Pemicu	Koordinator memilih untuk menghapus jadwal sidang
Aktor	Koordinator
Kondisi Awal	Koordinator telah login

Aliran: - Kejadian Normal	1. Koordinator melakukan aksi untuk membuka halaman kelola jadwal. 2. Koordinator menekan tombol <i>delete</i> jadwal sidang pada sidang yang akan diubah datanya. 3. Sistem menampilkan konfirmasi penghapusan jadwal sidang. 4. Koordinator melakukan aksi untuk memberikan konfirmasi penghapusan data. 5. Sistem menghapus jadwal sidang.
- Kejadian Alternatif	-
Kondisi Akhir	Jadwal sidang berhasil dihapus.



Powered By: Visual Paradigm Community Edition

Gambar 3.23 Diagram Aktivitas Menghapus Jadwal Sidang

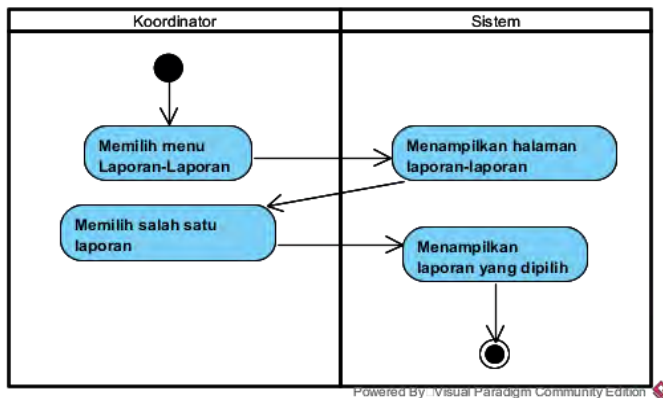
3.1.5.9 Melihat Laporan-Laporan

Laporan-laporan mengenai tugas akhir dibutuhkan koordinator untuk beberapa keperluan. Laporan-laporan tersebut dikumpulkan dalam sebuah menu untuk kemudahan aksesnya. Laporan bersifat statis, tidak ada aksi yang dapat dilakukan di halaman laporan. Berikut ini spesifikasi kasus penggunaan untuk

melihat laporan-laporan Tabel 3.19. Selain itu, Gambar 3.24 menunjukkan diagram aktivitas yang mendefinisikan aktivitas pengguna dan respon yang diberikan sistem.

Tabel 3.19 Spesifikasi Kasus Penggunaan Melihat Laporan-Laporan

Nama	Melihat laporan-laporan
Kode	FR-09
Deskripsi	Koordinator melihat laporan-laporan
Tipe	Fungsional
Pemicu	Koordinator memilih untuk melihat laporan-laporan
Aktor	Koordinator
Kondisi Awal	Koordinator telah login
Aliran: - Kejadian Normal	<ol style="list-style-type: none"> 1. Koordinator melakukan aksi untuk membuka menu laporan-laporan. 2. Sistem menampilkan rincian laporan-laporan yang dapat diakses. 3. Koordinator melakukan aksi untuk membuka salah satu laporan. 4. Sistem menampilkan laporan yang dipilih.
- Kejadian Alternatif	-
Kondisi Akhir	Laporan terpilih ditampilkan.



Gambar 3.24 Diagram Aktivitas Melihat Laporan-Laporan

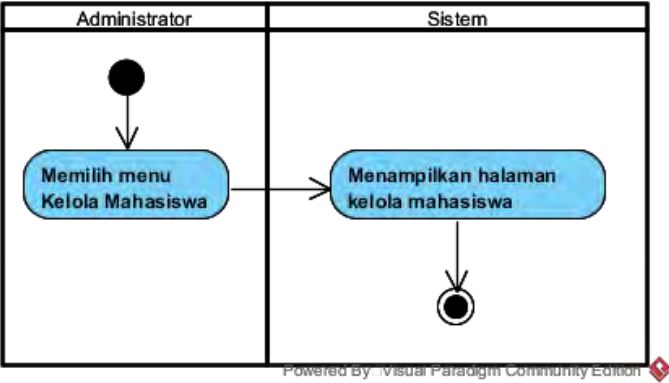
3.1.5.10 Mengelola Data Mahasiswa

Pengelolaan data mahasiswa dilakukan oleh administrator. Pengelolaan data meliputi penambahan, pengubahan, dan penghapusan. Spesifikasi-spesifikasi kasus penggunaan yang termasuk dalam pengelolaan data mahasiswa dapat dilihat pada Tabel 3.20 hingga Tabel 3.23. Selain itu, Gambar 3.25 hingga Gambar 3.28 menunjukkan diagram aktivitas yang mendefinisikan aktivitas pengguna dan respon yang diberikan sistem.

Tabel 3.20 Spesifikasi Kasus Penggunaan Melihat Daftar Mahasiswa

Nama	Melihat daftar mahasiswa
Kode	FR-10.1
Deskripsi	Administrator melihat daftar mahasiswa
Tipe	Fungsional
Pemicu	Administrator memilih untuk melihat daftar mahasiswa
Aktor	Administrator
Kondisi Awal	Administrator telah login

Aliran: - Kejadian Normal	1. Administrator melakukan aksi untuk membuka halaman daftar mahasiswa. 2. Sistem menampilkan halaman daftar mahasiswa.
- Kejadian Alternatif	-
Kondisi Akhir	Sistem menampilkan daftar mahasiswa.

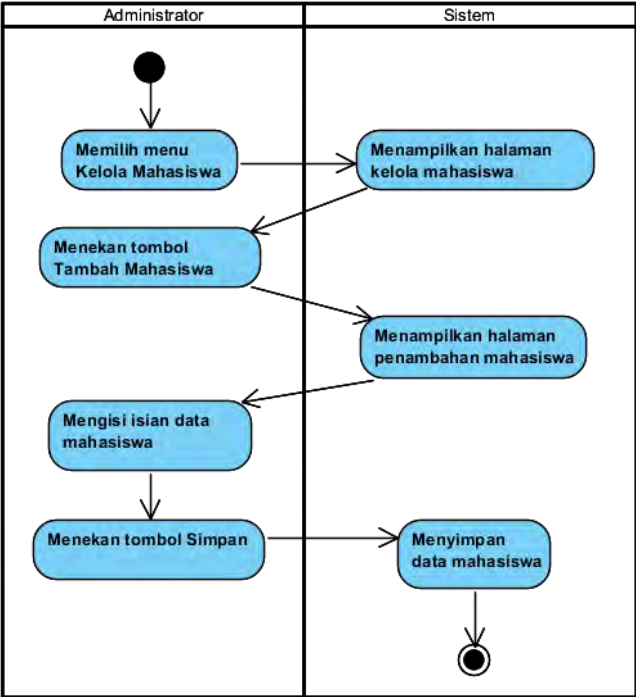


Gambar 3.25 Diagram Aktivitas Melihat Daftar Mahasiswa

Tabel 3.21 Spesifikasi Kasus Penggunaan Menambahkan Mahasiswa

Nama	Menambahkan mahasiswa
Kode	FR-10.2
Deskripsi	Administrator menambahkan mahasiswa
Tipe	Fungsional
Pemicu	Administrator memilih untuk menambahkan mahasiswa
Aktor	Administrator
Kondisi Awal	Administrator telah login

Aliran: - Kejadian Normal	1. Administrator melakukan aksi untuk membuka halaman penambahan mahasiswa. 2. Sistem menampilkan halaman penambahan mahasiswa. 3. Administrator mengisi isian data mahasiswa. 4. Administrator melakukan aksi untuk menyimpan data mahasiswa. 5. Sistem menyimpan data mahasiswa.
- Kejadian Alternatif	-
Kondisi Akhir	Mahasiswa baru berhasil ditambahkan.



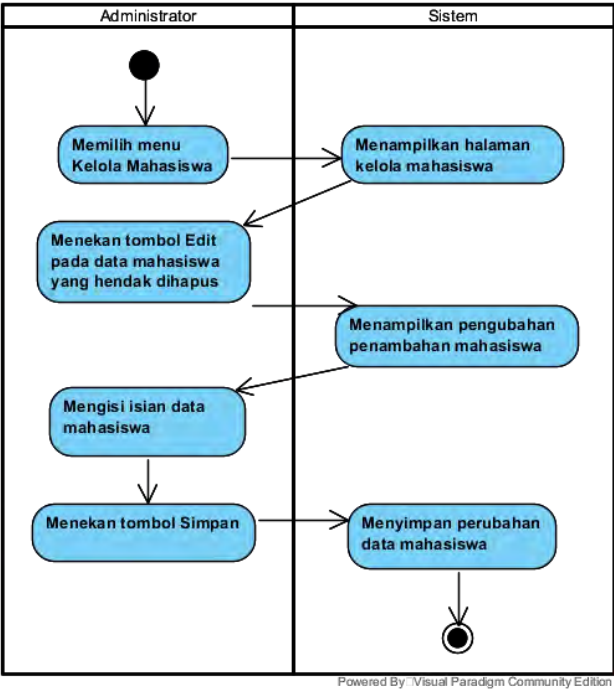
Powered By : Visual Paradigm Community Edition

Gambar 3.26 Diagram Aktivitas Menambahkan Mahasiswa

Tabel 3.22 Spesifikasi Kasus Penggunaan Mengubah Data Mahasiswa

Nama	Mengubah data mahasiswa
Kode	FR-10.3
Deskripsi	Administrator mengubah data mahasiswa
Tipe	Fungsional
Pemicu	Administrator memilih untuk mengubah data mahasiswa
Aktor	Administrator
Kondisi Awal	Administrator telah login

Aliran: - Kejadian Normal	<ol style="list-style-type: none"> 1. Administrator melakukan aksi untuk membuka halaman perubahan data mahasiswa. 2. Sistem menampilkan halaman perubahan data mahasiswa. 3. Administrator mengisi isian data mahasiswa pada bagian yang hendak diubah. 4. Administrator melakukan aksi untuk menyimpan perubahan data mahasiswa. 5. Sistem menyimpan perubahan data.
- Kejadian Alternatif	-
Kondisi Akhir	Data mahasiswa berhasil diubah.



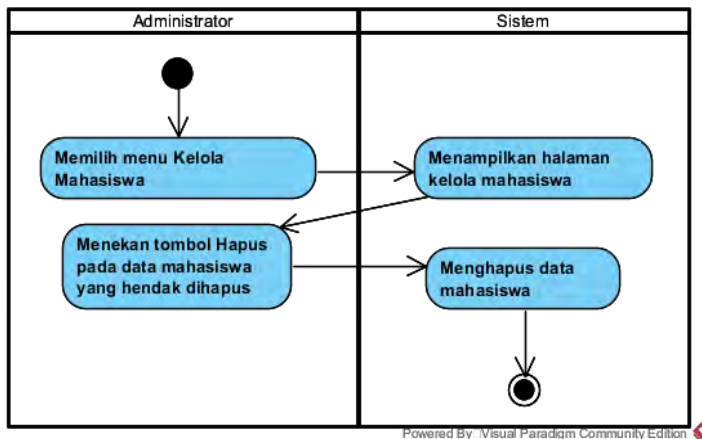
Powered By : Visual Paradigm Community Edition

Gambar 3.27 Diagram Aktivitas Mengubah Data Mahasiswa

Tabel 3.23. Spesifikasi Kasus Penggunaan Menghapus Mahasiswa

Nama	Menghapus mahasiswa
Kode	FR-10.4
Deskripsi	Administrator menghapus mahasiswa
Tipe	Fungsional
Pemicu	Administrator memilih untuk menghapus mahasiswa
Aktor	Administrator
Kondisi Awal	Administrator telah login
Aliran:	1. Administrator melakukan aksi untuk membuka halaman daftar mahasiswa.

- Kejadian Normal	2. Sistem menampilkan halaman daftar mahasiswa. 3. Administrator melakukan aksi untuk menghapus salah satu mahasiswa yang terdapat dalam daftar. 4. Sistem menampilkan pesan konfirmasi penghapusan mahasiswa. 5. Administrator melakukan aksi untuk memberi konfirmasi penghapusan. 6. Sistem menghapus data mahasiswa.
- Kejadian Alternatif	-
Kondisi Akhir	Mahasiswa dihapus.



Gambar 3.28 Diagram Aktivitas Menghapus Mahasiswa

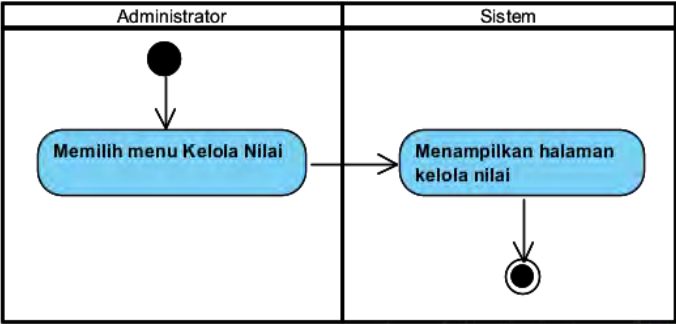
3.1.5.11 Mengelola Nilai Tugas Akhir

Nilai tugas akhir terdiri dari nilai sidang proposal dan nilai sidang akhir. Administrator berhak menambahkan nilai tugas akhir mahasiswa. Spesifikasi-spesifikasi kasus penggunaan pengelolaan nilai tugas akhir dapat dilihat pada Tabel 3.24 dan Tabel 3.25. Selain itu, Gambar 3.29 hingga Gambar 3.30 menunjukkan

diagram aktivitas yang mendefinisikan aktivitas pengguna dan respon yang diberikan sistem.

Tabel 3.24 Spesifikasi Kasus Penggunaan Melihat Daftar Nilai

Nama	Melihat daftar nilai
Kode	FR-11.1
Deskripsi	Administrator melihat daftar nilai tugas akhir seluruh mahasiswa
Tipe	Fungsional
Pemicu	Administrator memilih untuk melihat daftar nilai tugas akhir mahasiswa
Aktor	Administrator
Kondisi Awal	Administrator telah login
Aliran:	
- Kejadian Normal	1. Administrator melakukan aksi untuk membuka halaman daftar nilai. 2. Sistem menampilkan halaman daftar nilai tugas akhir mahasiswa.
- Kejadian Alternatif	-
Kondisi Akhir	Sistem menampilkan daftar nilai tugas akhir mahasiswa.

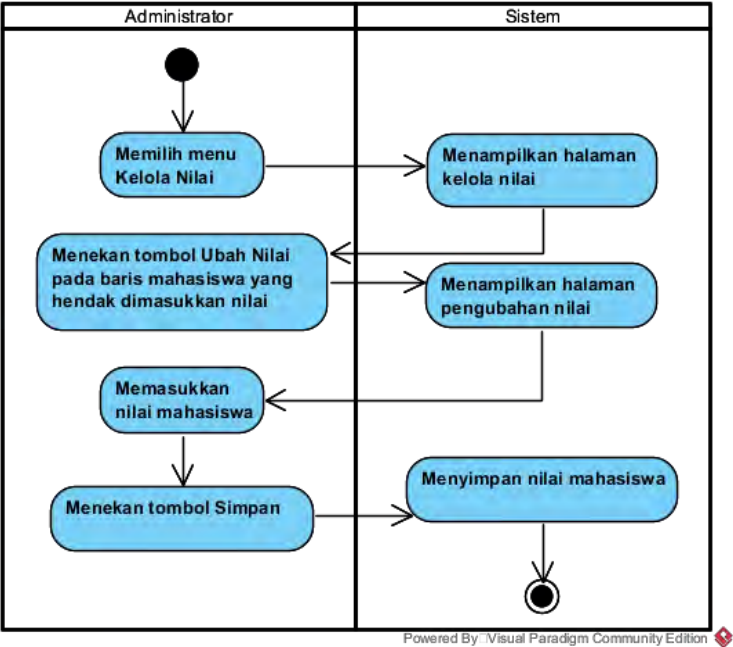


Powered By Visual Paradigm Community Edition

Gambar 3.29 Diagram Aktivitas Melihat Daftar Nilai

Tabel 3.25 Spesifikasi Kasus Penggunaan Memasukkan Nilai Tugas Akhir

Nama	Memasukkan nilai tugas akhir
Kode	FR-11.2
Deskripsi	Administrator memasukkan nilai tugas akhir mahasiswa, baik nilai sidang proposal maupun nilai sidang akhir
Tipe	Fungsional
Pemicu	Administrator memilih untuk memasukkan nilai tugas akhir
Aktor	Administrator
Kondisi Awal	Administrator telah login
Aliran: - Kejadian Normal	<ol style="list-style-type: none"> 1. Administrator melakukan aksi untuk membuka halaman daftar nilai tugas akhir. 2. Sistem menampilkan halaman daftar nilai tugas akhir mahasiswa. 3. Administrator melakukan aksi untuk melakukan pengisian nilai terhadap salah satu mahasiswa yang terdapat dalam daftar. 4. Sistem menampilkan halaman pengisian nilai tugas akhir untuk mahasiswa yang dipilih. 5. Administrator melakukan pengisian nilai tugas akhir. 6. Administrator melakukan aksi untuk menyimpan nilai. 7. Sistem menyimpan nilai tugas akhir mahasiswa.
- Kejadian Alternatif	-
Kondisi Akhir	Nilai tugas akhir mahasiswa berhasil disimpan.



Gambar 3.30 Diagram Aktivitas Memasukkan Nilai Tugas Akhir

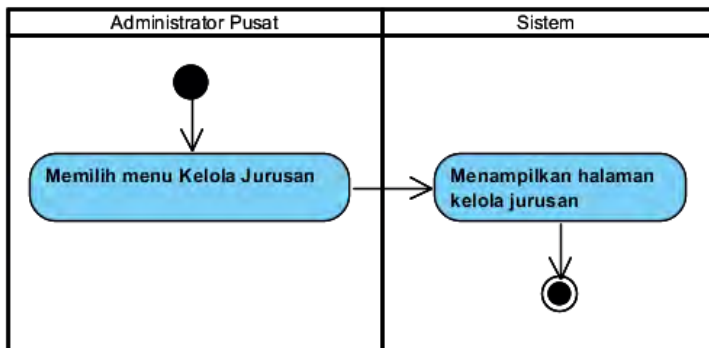
3.1.5.12 Mengelola Jurusan

Data jurusan merupakan salah satu data master yang dapat dikelola oleh administrator pusat terkait dengan pengaturan aliran kerja. Kasus penggunaan membuka halaman pengelolaan jurusan dapat dilihat pada Tabel 3.26. Selain itu, Gambar 3.31 menunjukkan diagram aktivitas yang mendefinisikan aktivitas pengguna dan respon yang diberikan sistem.

Tabel 3.26 Spesifikasi Kasus Penggunaan Melihat Halaman Pengelolaan Jurusan

Nama	Melihat halaman pengelolaan jurusan
Kode	FR-12

Deskripsi	Administrator melihat halaman pengelolaan jurusan
Tipe	Fungsional
Pemicu	Administrator memilih untuk melihat halaman pengelolaan jurusan
Aktor	Administrator Pusat
Kondisi Awal	Administrator telah login
Aliran:	1. Administrator melakukan aksi untuk membuka halaman pengeloaaan jurusan. 2. Sistem menampilkan halaman pengelolaan jurusan.
- Kejadian Normal	
- Kejadian Alternatif	-
Kondisi Akhir	Sistem menampilkan halaman pengelolaan jurusan.



Powered By : Visual Paradigm Community Edition

Gambar 3.31 Diagram Aktivitas Melihat Halaman Pengelolaan Jurusan

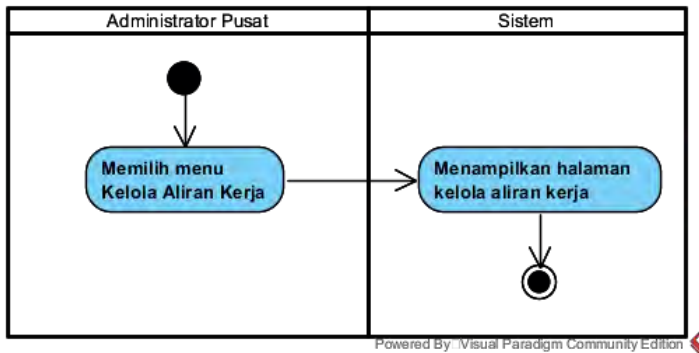
3.1.5.13 Mengelola Aliran Kerja

Data aliran kerja merupakan salah satu data master yang dapat dikelola oleh administrator. Kasus penggunaan membuka halaman pengelolaan aliran kerja dapat dilihat pada Tabel 3.27. Selain itu, Gambar 3.32 menunjukkan diagram aktivitas yang

mendefinisikan aktivitas pengguna dan respon yang diberikan sistem.

Tabel 3.27 Spesifikasi Kasus Penggunaan Melihat Halaman Pengelolaan Aliran Kerja

Nama	Melihat halaman pengelolaan aliran kerja
Kode	FR-13
Deskripsi	Administrator melihat halaman pengelolaan aliran kerja
Tipe	Fungsional
Pemicu	Administrator memilih untuk melihat halaman pengelolaan aliran kerja
Aktor	Administrator Pusat
Kondisi Awal	Administrator telah login
Aliran: - Kejadian Normal	1. Administrator melakukan aksi untuk membuka halaman pengelolaan aliran kerja. 2. Sistem menampilkan halaman pengelolaan aliran kerja.
- Kejadian Alternatif	-
Kondisi Akhir	Sistem menampilkan halaman pengelolaan aliran kerja.



Gambar 3.32 Diagram Aktivitas Melihat Halaman Pengelolaan Aliran Kerja

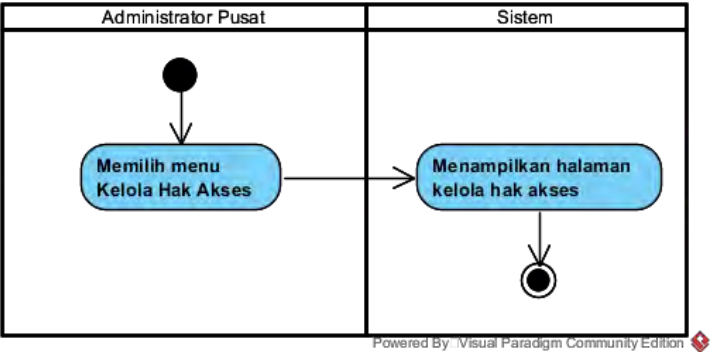
3.1.5.14 Mengelola Hak Akses

Data hak akses merupakan salah satu data master yang dapat dikelola oleh administrator pusat terkait dengan pengaturan aliran kerja. Kasus penggunaan membuka halaman pengelolaan hak akses dapat dilihat pada Tabel 3.28. Selain itu, Gambar 3.33 menunjukkan diagram aktivitas yang mendefinisikan aktivitas pengguna dan respon yang diberikan sistem.

Tabel 3.28 Spesifikasi Kasus Penggunaan Melihat Halaman Pengelolaan Hak Akses

Nama	Melihat halaman pengelolaan hak akses
Kode	FR-14
Deskripsi	Administrator melihat halaman pengelolaan hak akses
Tipe	Fungsional
Pemicu	Administrator memilih untuk melihat halaman pengelolaan hak akses
Aktor	Administrator Pusat
Kondisi Awal	Administrator telah login

Aliran: - Kejadian Normal	1. Administrator melakukan aksi untuk membuka halaman pengeloaan hak akses. 2. Sistem menampilkan halaman pengelolaan hak akses.
- Kejadian Alternatif	-
Kondisi Akhir	Sistem menampilkan halaman pengelolaan hak akses.



Gambar 3.33 Diagram Aktivitas Melihat Halaman Pengelolaan Hak Akses

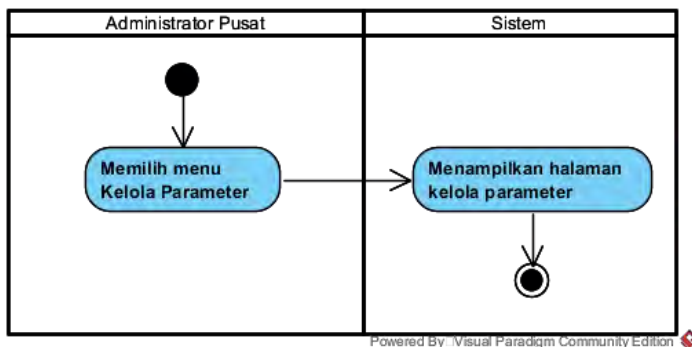
3.1.5.15 Mengelola Parameter

Data parameter merupakan salah satu data master yang dapat dikelola oleh administrator pusat terkait dengan pengaturan aliran kerja. Kasus penggunaan membuka halaman pengelolaan parameter dapat dilihat pada Tabel 3.29. Selain itu, Gambar 3.34 menunjukkan diagram aktivitas yang mendefinisikan aktivitas pengguna dan respon yang diberikan sistem.

Tabel 3.29 Spesifikasi Kasus Penggunaan Melihat Halaman Pengelolaan Parameter

Nama	Melihat halaman pengelolaan parameter
Kode	FR-15

Deskripsi	Administrator melihat halaman pengelolaan parameter
Tipe	Fungsional
Pemicu	Administrator memilih untuk melihat halaman pengelolaan parameter
Aktor	Administrator Pusat
Kondisi Awal	Administrator telah login
Aliran:	1. Administrator melakukan aksi untuk membuka halaman pengeloaan parameter. 2. Sistem menampilkan halaman pengelolaan parameter.
- Kejadian Normal	
- Kejadian Alternatif	-
Kondisi Akhir	Sistem menampilkan halaman pengelolaan parameter.



Gambar 3.34 Diagram Aktivitas Melihat Halaman Pengelolaan Parameter

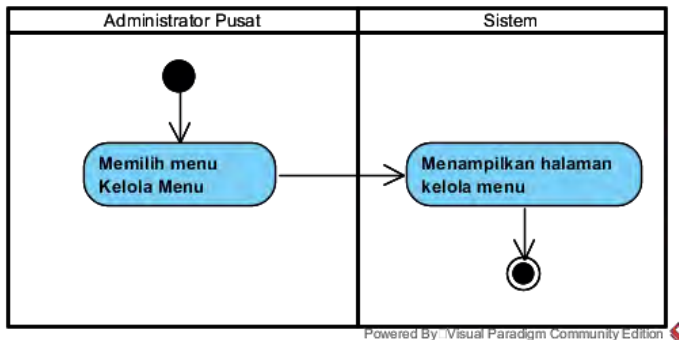
3.1.5.16 Mengelola Menu

Data menu merupakan salah satu data master yang dapat dikelola oleh administrator pusat terkait dengan pengaturan aliran kerja. Kasus penggunaan membuka halaman pengelolaan menu dapat dilihat pada Tabel 3.30. Selain itu, Gambar 3.35

menunjukkan diagram aktivitas yang mendefinisikan aktivitas pengguna dan respon yang diberikan sistem.

Tabel 3.30 Spesifikasi Kasus Penggunaan Melihat Halaman Pengelolaan Menu

Nama	Melihat halaman pengelolaan menu
Kode	FR-16
Deskripsi	Administrator melihat halaman pengelolaan menu
Tipe	Fungsional
Pemicu	Administrator memilih untuk melihat halaman pengelolaan menu
Aktor	Administrator Pusat
Kondisi Awal	Administrator telah login
Aliran: - Kejadian Normal	1. Administrator melakukan aksi untuk membuka halaman pengelolaan menu. 2. Sistem menampilkan halaman pengelolaan menu.
- Kejadian Alternatif	-
Kondisi Akhir	Sistem menampilkan halaman pengelolaan menu.



Gambar 3.35 Diagram Aktivitas Melihat Halaman Pengelolaan Menu

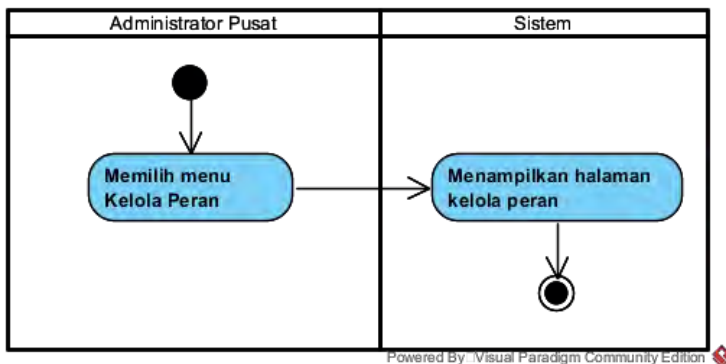
3.1.5.17 Mengelola Peran

Data peran merupakan salah satu data master yang dapat dikelola oleh administrator pusat terkait dengan pengaturan aliran kerja. Kasus penggunaan membuka halaman pengelolaan peran dapat dilihat pada Tabel 3.31. Selain itu, Gambar 3.36 menunjukkan diagram aktivitas yang mendefinisikan aktivitas pengguna dan respon yang diberikan sistem.

Tabel 3.31 Spesifikasi Kasus Penggunaan Melihat Halaman Pengelolaan Peran

Nama	Melihat halaman pengelolaan peran
Kode	FR-17
Deskripsi	Administrator melihat halaman pengelolaan peran
Tipe	Fungsional
Pemicu	Administrator memilih untuk melihat halaman pengelolaan peran
Aktor	Administrator Pusat
Kondisi Awal	Administrator telah login
Aliran:	1. Administrator melakukan aksi untuk membuka halaman pengeloan peran.

- Kejadian Normal	2. Sistem menampilkan halaman pengelolaan peran.
- Kejadian Alternatif	-
Kondisi Akhir	Sistem menampilkan halaman pengelolaan peran.

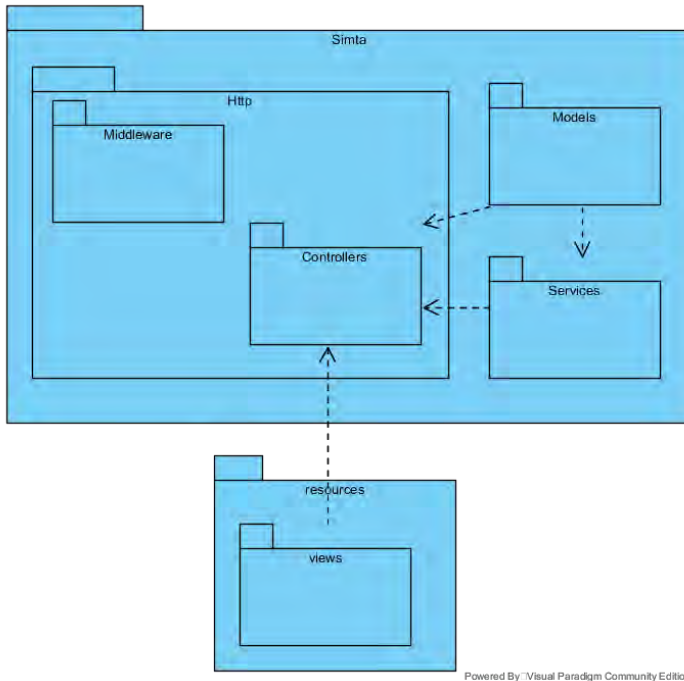


Gambar 3.36 Diagram Aktivitas Melihat Halaman Pengelolaan Peran

3.2 Perancangan Sistem

Penjelasan tahap perancangan perangkat lunak dibagi menjadi tiga bagian yaitu perancangan arsitektur sistem, perancangan diagram basis data, dan perancangan antarmuka.

3.2.1 Perancangan Arsitektur Perangkat Lunak



Gambar 3.37 Arsitektur Perangkat Lunak

Perangkat lunak yang dibangun menerapkan pola perancangan MVC (*Model-View-Controller*). Gambar 3.37 menunjukkan arsitektur perangkat lunak yang diterapkan. Kelas-kelas berada dalam ruang nama (*namespace*) *Simta*. Sedangkan, kode program tampilan berada dalam direktori *resources/views*. Controller merupakan bagian utama yang mengatur proses bisnis sistem. Beberapa controller memanfaatkan komponen dalam paket *Services* sebagai penampung kode proses bisnis. Sedangkan, beberapa controller lainnya mengimplementasikan logika bisnis langsung di dalam *method*-nya sehingga harus memanfaatkan

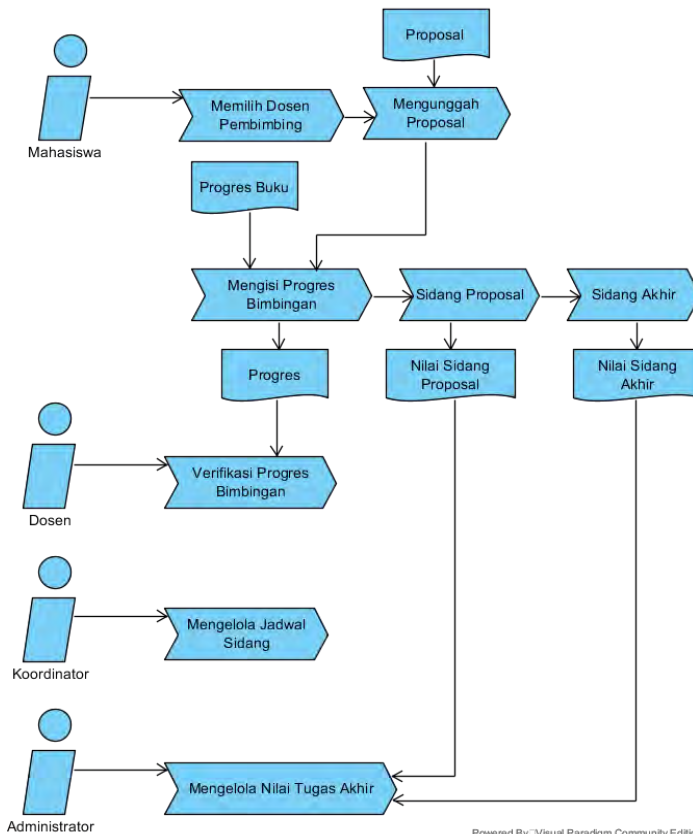
langsung komponen *Models* untuk kebutuhan penulisan dan pembacaan data.

3.2.2 Perancangan Proses Aplikasi

Pada subbab ini dijelaskan mengenai rancangan proses aplikasi yang akan diimplementasikan untuk menyelesaikan permasalahan yang telah diidentifikasi.

3.2.2.1 Rancangan Proses Sistem Generik

Sistem informasi administrasi tugas akhir yang bersifat generik adalah sistem yang dapat digunakan oleh berbagai jurusan untuk melaksanakan prosedur tugas akhir di jurusan. Sistem generik tersebut harus dapat menangani proses bisnis utama dari berbagai macam prosedur atau aliran kerja pelaksanaan tugas akhir yang ada. Selain itu, sistem juga harus dapat diatur sedemikian rupa sehingga perubahan prosedur suatu jurusan dapat dieksekusi secara sederhana.



Gambar 3.38 Aliran Kerja Utama SIMTA ITS

Aliran kerja utama dalam pelaksanaan tugas akhir di ITS diperoleh berdasarkan hasil analisis kebutuhan dari beberapa jurusan. Aliran kerja utama sistem ditunjukkan pada Gambar 3.38. Aliran kerja tersebut diperoleh berdasarkan hasil analisis kesamaan proses dari beberapa prosedur tugas akhir yang telah dijelaskan pada bab 2.1. Berikut ini penjelasan tentang setiap proses dan relasinya dengan proses-proses pada setiap prosedur tugas akhir dari lima jurusan berbeda.

1. Memilih Dosen Pembimbing

Pemilihan dosen pembimbing menjadi proses generik karena dilakukan proses ini juga terdapat dalam kelima prosedur yang telah dijelaskan sebelumnya. Meskipun demikian, terdapat perbedaan jumlah dosen yang harus dipilih oleh mahasiswa. Perbedaan tersebut ditampung dalam parameter sistem untuk dapat diatur oleh setiap jurusan.

2. Mengunggah Proposal

Proposal merupakan entitas yang muncul di setiap prosedur tugas akhir kelima jurusan meskipun dengan istilah yang berbeda. Jurusan Kimia menyebut proposal sebagai rancangan tugas akhir, sedangkan Jurusan Teknik Sistem Perkapalan menggunakan istilah proposal untuk proposal pengajuan judul yang dipresentasikan dalam presentasi judul (P1). Proses pengunggahan proposal juga disertai proses pengisian judul dan abstraksi tugas akhir.

3. Mengisi Progres Bimbingan

Progres bimbingan merupakan proses yang mewakili proses pengerjaan tugas akhir pada kelima jurusan. Mahasiswa di setiap jurusan dapat memasukkan rangkuman progres ke dalam sistem setiap kali dilakukan progres bimbingan ke dosen pembimbing. Data yang tersimpan dalam proses ini dapat digunakan oleh proses lainnya bahkan pada proses spesifik yang tidak terdapat dalam aliran kerja sistem generik.

4. Sidang Proposal dan Sidang Akhir

Sidang proposal dan sidang akhir merupakan proses yang dapat mewakili proses-proses di jurusan yang bersifat pengujian terhadap tugas akhir mahasiswa. Sidang proposal dapat mewakili proses ujian proposal pada Jurusan Teknik Geomatika dan proses presentasi judul pada Jurusan Teknik Perkapalan. Sedangkan, sidang akhir dapat mewakili ujian paling akhir untuk tugas akhir pada setiap jurusan.

5. Verifikasi Progres Bimbingan

Verifikasi progres bimbingan merupakan proses yang berpasangan dengan proses mengisi progres bimbingan. Kedua

proses tersebut ada untuk mencukupi proses pemantauan pada proses pengerjaan tugas akhir.

6. Mengelola Sidang

Pengelolaan sidang dibutuhkan oleh koordinator tugas akhir jurusan untuk mengatur jalannya proses pengujian tugas akhir. Melalui fitur pengelolaan sidang, koordinator jurusan dapat mengatur kegiatan pengujian sesuai dengan pengujian yang ada di jurusannya masing-masing.

7. Mengelola Nilai

Pengelolaan nilai merupakan proses yang dibutuhkan sebagai akhir dari proses pengujian tugas akhir mahasiswa. Proses ini diserahkan kepada administrator jurusan yang bertanggung jawab mengelola perihal administrasi penilaian mahasiswa.

3.2.2.2 Rancangan Proses Pengaturan Aliran Kerja

Beberapa jurusan di ITS memiliki proses unik yang tidak terdapat dalam aliran kerja utama sebagaimana dijelaskan pada subbab sebelumnya. Misalkan, MonTA jurusan Teknik Informatika menyediakan fitur cetak sampul proposal dan lembar persetujuan maju sidang. Contoh lainnya adalah jurusan Teknik Mesin menerapkan sistem antrian dalam proses penentuan dosen pembimbing.

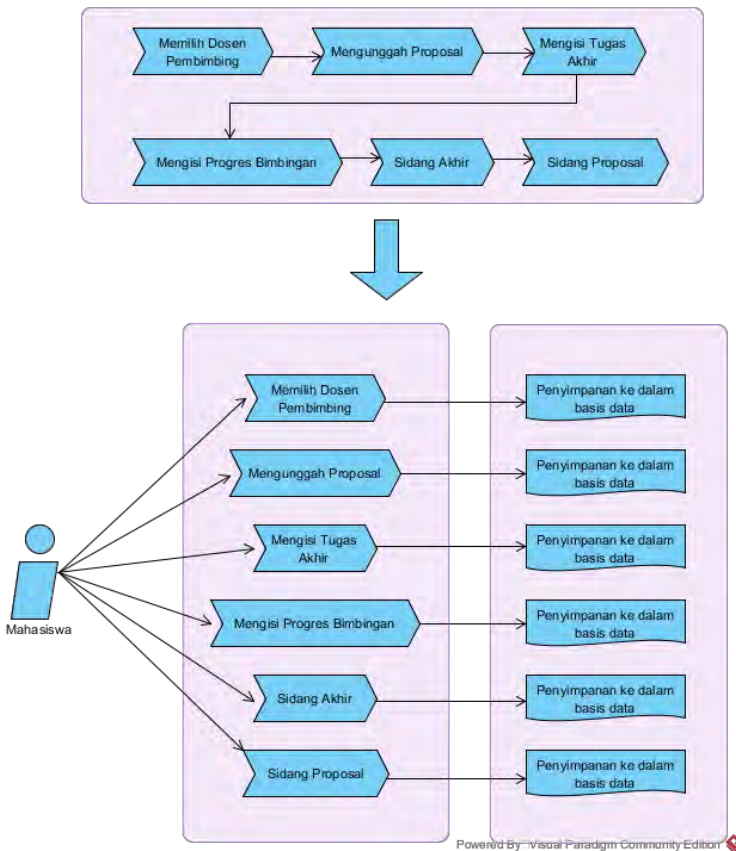
Untuk memenuhi kebutuhan tersebut, sistem harus menyediakan sebuah fitur pengaturan di mana aliran kerja prosedur tugas akhir sebuah jurusan dapat ditentukan secara fleksibel. Dalam fitur tersebut, masing-masing jurusan ditentukan aliran kerjanya sehingga pengguna di jurusan tersebut dapat mengakses fitur-fitur yang sesuai dengan prosedur yang harus diikuti.

Sebuah fitur dapat diakses dalam sistem melalui sebuah menu. Setiap jenis pengguna dalam sistem memiliki menu yang berbeda-beda sesuai dengan peran dan kebutuhannya. Melalui menu-menu tersebutlah pengguna mahasiswa, misalnya, menjalankan proses-proses yang harus dilalui dalam prosedur mengerjakan tugas akhir.

Jenis atau peran pengguna antara jurusan satu dengan lainnya dapat berbeda. Hal ini disesuaikan dengan prosedur tugas akhir di jurusan masing-masing. Satu jenis pengguna dengan nama peran yang sama bisa jadi memiliki hak akses dan menu yang berbeda.

Berdasarkan keterangan tersebut, fitur pengaturan aliran kerja prosedur tugas akhir dapat diwujudkan dengan pemaketan menu dan hak akses pengguna. Paket tersebut memetakan antara peran atau jenis pengguna terhadap menu dan hak akses yang bisa dieksekusi. Pemetaan menu dan hak akses disesuaikan dengan prosedur tugas akhir di jurusan.

Pemaketan menu dan hak akses didasarkan pada bentuk atomik dari aliran kerja prosedur tugas akhir. Dua proses yang saling berhubungan dalam sebuah aliran kerja dapat diimplementasikan sebagai dua fitur atau halaman yang berbeda yang dihubungkan dengan hasil sementara. Fitur atau halaman dapat diakses melalui sebuah menu. Sedangkan, hasil sementara merupakan keluaran dari proses pertama dan merupakan masukan dari proses kedua. Dengan demikian, keseluruhan aliran kerja untuk setiap jenis pengguna dapat disajikan dalam bentuk paket menu dan hak akses yang merujuk pada fitur atau halaman sesuai dengan proses yang harus dilalui oleh pengguna tersebut. Gambar 3.39 menunjukkan proses pemaketan aliran kerja untuk pengguna mahasiswa dalam sistem generik.



Gambar 3.39 Pemaketan Aliran Kerja Prosedur Tugas Akhir Bagi Pengguna Mahasiswa

Perubahan prosedur tugas akhir di suatu jurusan akan diikuti dengan perubahan paket yang ada di sistem. Jika perubahan tersebut melibatkan sebuah fitur baru, menu baru harus ditambahkan ke dalam paket setelah fitur tersebut diimplementasikan. Perubahan paket tersebut dapat dilakukan melalui fitur pengelolaan paket.

3.2.2.3 Rancangan Proses Pengaturan Parameter

Parameter aliran kerja adalah nilai-nilai yang dijadikan acuan dalam proses. Parameter bersifat fleksibel karena dapat berubah sewaktu-waktu mengikuti kebijakan yang ada. Contoh parameter aliran kerja dalam prosedur pelaksanaan tugas akhir adalah jumlah dosen pembimbing untuk tiap tugas akhir mahasiswa.

Pengaturan parameter aliran kerja dapat dilakukan oleh koordinator tugas akhir jurusan. Fitur pengaturan parameter aliran kerja merupakan sebuah fitur di mana seluruh parameter dapat dikelola dalam sebuah halaman web.

Sebuah parameter terdiri dari kode, label, nilai, dan keterangan parameter. Kode parameter adalah identitas unik sebuah parameter yang dirujuk langsung oleh kode program. Label merupakan informasi yang ditampilkan kepada pengguna sebagai nama dari parameter. Keterangan merupakan penjelasan mengenai parameter. Sedangkan, nilai merupakan nilai parameter.

3.2.3 Perancangan Basis Data

Pada subbab ini dijelaskan tentang rancangan basis data yang akan digunakan oleh sistem. Rancangan basis data dibuat langsung dalam bentuk model data fisik. Hal tersebut membuat proses perancangan menjadi relatif lebih mudah bagi penulis dan lebih dekat dengan implementasi dalam kerangka kerja pemrograman Laravel yang memanfaatkan migrasi dalam bentuk skema sebagai sarana untuk mengelola basis data. Diagram model data fisik sistem ditunjukkan oleh Gambar 3.40.

Basis data akan dibangun menggunakan RDBMS MySQL. MySQL merupakan RDBMS yang tersedia secara gratis, memiliki proses pemasangan yang mudah dan cepat, dan dikenal oleh sebagian besar pengembang perangkat lunak.

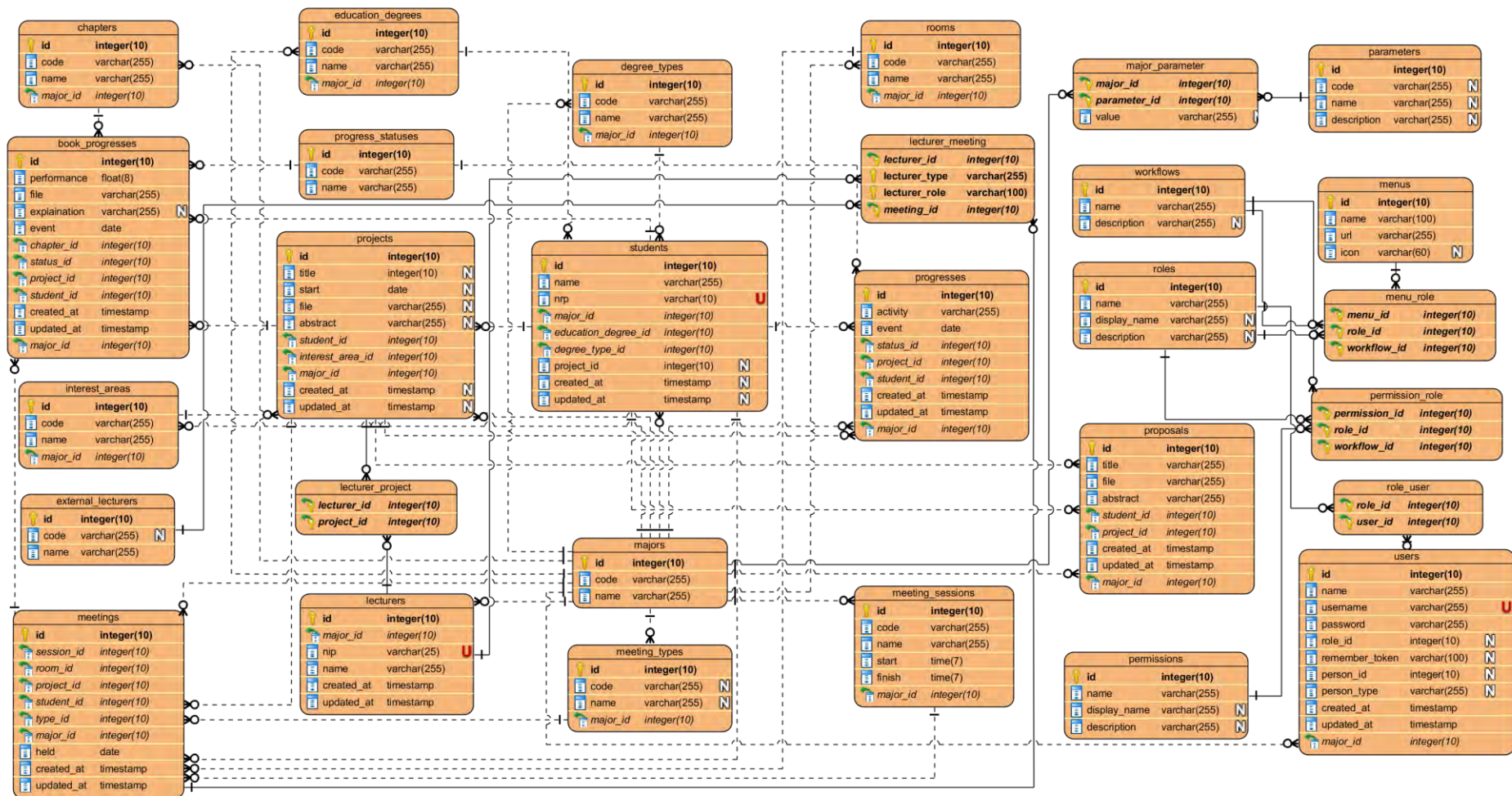
Konvensi yang diterapkan pada perancangan basis data ini mengikuti konvensi yang diterapkan oleh Laravel. Nama tabel dan kolom diberi nama dalam bahasa Inggris. *Primary key* untuk tiap tabel diberinama 'id'. Sedangkan, *foreign key* diberinama dengan format nama tabel rujukan ditambah kata kunci 'id'.

Rincian rancangan masing-masing tabel dijelaskan pada tabel tersendiri. Berikut ini rincian rancangan tabel yang terdapat dalam basis data.

Tabel 3.32 menunjukkan rancangan tabel *chapters*. Tabel tersebut menyimpan informasi bab-bab yang menjadi pilihan saat mahasiswa memasukkan progres buku bimbingan.

Tabel 3.32 Rancangan Tabel *Chapters*

Nama Kolom	Keterangan
ID	<i>Primary key</i>
MAJOR_ID	<i>Foreign key</i> yang merujuk ke tabel <i>Majors</i>
CODE	Singkatan atau kode bab
NAME	Nama bab



Gambar 3.40 Model Data Fisik

Tabel 3.33 menunjukkan rancangan tabel *book_progresses*. Tabel tersebut menyimpan semua progres buku yang dilaporkan oleh mahasiswa.

Tabel 3.33 Rancangan Tabel *Book Progresses*

Nama Kolom	Keterangan
ID	<i>Primary key</i>
MAJOR_ID	<i>Foreign key</i> yang merujuk ke tabel <i>Majors</i>
CHAPTER_ID	<i>Foreign key</i> yang merujuk ke tabel <i>Chapters</i>
STATUS_ID	<i>Foreign key</i> yang merujuk ke tabel <i>Progress Statuses</i>
PROJECT_ID	<i>Foreign key</i> yang merujuk ke tabel <i>Projects</i>
STUDENT_ID	<i>Foreign key</i> yang merujuk ke tabel <i>Students</i>
PERFORMANCE	Pencapaian mahasiswa dalam menyelesaikan bab tertentu
FILE	Lokasi berkas yang diunggah
EXPLAINATION	Penjelasan mengenai progres buku
EVENT	Tanggal progres dilakukan
CREATED_AT	Tanggal progres pertama kali disimpan
UPDATED_AT	Tanggal progres terakhir kali diperbarui

Tabel 3.34 menunjukkan rancangan tabel *interest areas*. Tabel tersebut menyimpan data laboratorium atau rumpun mata kuliah.

Tabel 3.34. Rancangan Tabel *Interest Areas*

Nama Kolom	Keterangan
ID	<i>Primary key</i>
MAJOR_ID	<i>Foreign key</i> yang merujuk ke tabel <i>Majors</i>
CODE	Singkatan atau kode laboratorium
NAME	Nama laboratorium

Tabel 3.35 menunjukkan rancangan tabel *education degrees*. Tabel tersebut menyimpan data jenjang pendidikan yang tersedia di masing-masing jurusan.

Tabel 3.35 Rancangan Tabel *Education Degrees*

Nama Kolom	Keterangan
-------------------	-------------------

ID	<i>Primary key</i>
MAJOR_ID	<i>Foreign key</i> yang merujuk ke tabel <i>Majors</i>
CODE	Singkatan atau kode jenjang pendidikan
NAME	Nama jenjang pendidikan

Tabel 3.36 menunjukkan rancangan tabel *degree types*. Tabel tersebut menyimpan jenis jenjang pendidikan yang tersedia di masing-masing jurusan.

Tabel 3.36 Rancangan Tabel *Degree Types*

Nama Kolom	Keterangan
ID	<i>Primary key</i>
MAJOR_ID	<i>Foreign key</i> yang merujuk ke tabel <i>Majors</i>
CODE	Singkatan atau kode jenjang pendidikan
NAME	Nama jenjang pendidikan

Tabel 3.37 menunjukkan rancangan tabel *meeting types*. Tabel tersebut menyimpan jenis tipe sidang.

Tabel 3.37 Rancangan Tabel *Meeting Types*

Nama Kolom	Keterangan
ID	<i>Primary key</i>
MAJOR_ID	<i>Foreign key</i> yang merujuk ke tabel <i>Majors</i>
CODE	Singkatan atau kode jenis sidang
NAME	Nama jenis sidang

Tabel 3.38 menunjukkan rancangan tabel *rooms*. Tabel tersebut menyimpan data ruangan yang digunakan untuk sidang.

Tabel 3.38 Rancangan Tabel *Rooms*

Nama Kolom	Keterangan
ID	<i>Primary key</i>
MAJOR_ID	<i>Foreign key</i> yang merujuk ke tabel <i>Majors</i>
CODE	Singkatan atau kode ruangan
NAME	Nama ruangan

Tabel 3.39 menunjukkan rancangan tabel *majors*. Tabel tersebut menyimpan nama-nama jurusan.

Tabel 3.39 Rancangan Tabel *Majors*

Nama Kolom	Keterangan
------------	------------

ID	<i>Primary key</i>
CODE	Singkatan atau kode jurusan
NAME	Nama jurusan

Tabel 3.40 menunjukkan rancangan tabel *external lecturers*. Tabel tersebut menyimpan nama-nama dosen penguji yang berasal dari luar jurusan.

Tabel 3.40 Rancangan Tabel *External Lecturers*

Nama Kolom	Keterangan
ID	<i>Primary key</i>
CODE	ID atau NIP dosen
NAME	Nama dosen

Tabel 3.41 menunjukkan rancangan tabel *meetings*. Tabel tersebut menyimpan jadwal sidang, baik sidang proposal maupun sidang akhir.

Tabel 3.41 Rancangan Tabel *Meetings*

Nama Kolom	Keterangan
ID	<i>Primary key</i>
MAJOR_ID	<i>Foreign key</i> yang merujuk ke tabel <i>Majors</i>
SESSION_ID	<i>Foreign key</i> yang merujuk ke tabel <i>Meeting Sessions</i>
ROOM_ID	<i>Foreign key</i> yang merujuk ke tabel <i>Rooms</i>
PROJECT_ID	<i>Foreign key</i> yang merujuk ke tabel <i>Projects</i>
STUDENT_ID	<i>Foreign key</i> yang merujuk ke tabel <i>Students</i>
TYPE_ID	<i>Foreign key</i> yang merujuk ke tabel <i>Meeting Types</i> .
HELD	Tanggal sidang
CREATED_AT	Tanggal jadwal sidang pertama kali disimpan
UPDATED_AT	Tanggal jadwal sidang terakhir diubah

Tabel 3.42 menunjukkan rancangan tabel *progress statuses*. Tabel tersebut merupakan tabel referensi status progres bimbingan.

Tabel 3.42 Rancangan Tabel *Progress Statuses*

Nama Kolom	Keterangan
-------------------	-------------------

ID	<i>Primary key</i>
CODE	Singkatan atau kode status progres
NAME	Nama status progres

Tabel 3.43 menunjukkan rancangan tabel *projects*. Tabel tersebut menyimpan data tugas akhir mahasiswa.

Tabel 3.43 Rancangan Tabel *Projects*

Nama Kolom	Keterangan
ID	<i>Primary key</i>
TITLE	Judul tugas akhir
START	Tanggal tugas akhir mulai dikerjakan
FILE	Lokasi berkas tugas akhir yang diunggah
ABSTRACT	Abstraksi tugas akhir
STUDENT_ID	<i>Foreign key</i> yang merujuk ke tabel <i>Students</i>
INTEREST_AREA_ID	<i>Foreign key</i> yang merujuk ke tabel <i>Interest Areas</i>
MAJOR_ID	<i>Foreign key</i> yang merujuk ke tabel <i>Majors</i>
CREATED_AT	Tanggal pertama kali disimpan
UPDATED_AT	Tanggal terakhir diubah

Tabel 3.44 menunjukkan rancangan tabel *proposals*. Tabel tersebut menyimpan data proposal mahasiswa.

Tabel 3.44 Rancangan Tabel *Proposals*

Nama Kolom	Keterangan
ID	<i>Primary key</i>
TITLE	Judul proposal tugas akhir
FILE	Lokasi berkas proposal yang diunggah
ABSTRACT	Abstraksi proposal tugas akhir
STUDENT_ID	<i>Foreign key</i> yang merujuk ke tabel <i>Students</i>
PROJECT_ID	<i>Foreign key</i> yang merujuk ke tabel <i>Projects</i>
MAJOR_ID	<i>Foreign key</i> yang merujuk ke tabel <i>Majors</i>
CREATED_AT	Tanggal pertama kali disimpan
UPDATED_AT	Tanggal terakhir diubah

Tabel 3.45 menunjukkan rancangan tabel *lecturer project*. Tabel tersebut merupakan tabel *pivot* antara tabel *lecturers* dan *projects* yang menyimpan rincian dosen pembimbing tugas akhir.

Tabel 3.45 Rancangan Tabel *Lecturer Project*

Nama Kolom	Keterangan
LECTURER_ID	<i>Primary key</i> sekaligus <i>foreign key</i> yang merujuk ke tabel <i>Lecturers</i>
PROJECT_ID	<i>Primary key</i> sekaligus <i>foreign key</i> yang merujuk ke tabel <i>Projects</i>

Tabel 3.46 menunjukkan rancangan tabel *lecturers*. Tabel tersebut menyimpan data dosen.

Tabel 3.46 Rancangan Tabel *Lecturers*

Nama Kolom	Keterangan
ID	<i>Primary key</i>
NIP	NIP dosen
NAME	Nama dosen
MAJOR_ID	<i>Foreign key</i> yang merujuk ke tabel <i>Majors</i>
CREATED_AT	Tanggal pertama kali disimpan
UPDATED_AT	Tanggal terakhir diubah

Tabel 3.47 menunjukkan rancangan tabel *students*. Tabel tersebut menyimpan data mahasiswa.

Tabel 3.47 Rancangan Tabel *Students*

Nama Kolom	Keterangan
ID	<i>Primary key</i>
NRP	NRP mahasiswa
NAME	Nama mahasiswa
MAJOR_ID	<i>Foreign key</i> yang merujuk ke tabel <i>Majors</i>
EDUCATION_DEGREE_ID	<i>Foreign key</i> yang merujuk ke tabel <i>Education Degrees</i>
DEGREE_TYPE_ID	<i>Foreign key</i> yang merujuk ke tabel <i>Degree Types</i>
PROJECT_ID	ID tugas akhir mahasiswa yang sedang aktif. Kolom ini tidak memiliki <i>foreign key</i> karena terdapat kondisi di mana

	kolom bernilai null, yaitu saat inisialisasi
CREATED_AT	Tanggal pertama kali disimpan
UPDATED_AT	Tanggal terakhir diubah

Tabel 3.48 menunjukkan rancangan tabel *lecturer meeting*. Tabel tersebut merupakan tabel pivot antara tabel *lecturers* dan *meetings* yang menyimpan rincian dosen penguji sidang.

Tabel 3.48 Rancangan Tabel *Lecturer Meeting*

Nama Kolom	Keterangan
LECTURER_ID	<i>Primary key</i> dan ID yang merujuk ke tabel <i>Lecturer</i> atau ke tabel <i>External Lecturers</i> .
LECTURER_TYPE	<i>Primary key</i> dan informasi apakah ID pada LECTURER_ID berasal dari tabel <i>Lecturers</i> atau <i>External Lecturers</i>
LECTURER_ROLE	<i>Primary key</i> , menyatakan apakah dosen penguji merupakan dosen pembimbing
MEETING_ID	<i>Primary key, foreign key</i> yang merujuk ke tabel <i>Meetings</i>

Tabel 3.49 menunjukkan rancangan tabel *progresses*. Tabel tersebut menyimpan semua progres bimbingan yang dilaporkan oleh mahasiswa.

Tabel 3.49 Rancangan Tabel *Progresses*

Nama Kolom	Keterangan
ID	<i>Primary key</i>
MAJOR_ID	<i>Foreign key</i> yang merujuk ke tabel <i>Majors</i>
STATUS_ID	<i>Foreign key</i> yang merujuk ke tabel <i>Progress Statuses</i>
PROJECT_ID	<i>Foreign key</i> yang merujuk ke tabel <i>Projects</i>
STUDENT_ID	<i>Foreign key</i> yang merujuk ke tabel <i>Students</i>
ACTIVITY	Penjelasan mengenai aktivitas progres bimbingan
EVENT	Tanggal progres dilakukan
CREATED_AT	Tanggal progres pertama kali disimpan
UPDATED_AT	Tanggal progres terakhir kali diperbarui

Tabel 3.50 menunjukkan rancangan tabel *roles*. Tabel tersebut menyimpan jenis-jenis hak akses pengguna.

Tabel 3.50 Rancangan Tabel *Roles*

Nama Kolom	Keterangan
ID	<i>Primary key</i>
NAME	Nama hak akses yang digunakan langsung dalam kode program
DISPLAY_NAME	Nama hak akses yang ditampilkan dalam halaman
DESCRIPTION	Deskripsi hak akses

Tabel 3.51 menunjukkan rancangan tabel *permissions*. Tabel tersebut menyimpan izin akses ke halaman tertentu.

Tabel 3.51 Rancangan Tabel *Permissions*

Nama Kolom	Keterangan
ID	<i>Primary key</i>
NAME	Nama ijin akses halaman yang digunakan langsung dalam kode program
DISPLAY_NAME	Nama ijin akses halaman yang ditampilkan dalam halaman
DESCRIPTION	Deskripsi ijin akses halaman

Tabel 3.52 menunjukkan rancangan tabel *users*. Tabel tersebut menyimpan akun pengguna.

Tabel 3.52 Rancangan Tabel *Users*

Nama Kolom	Keterangan
ID	<i>Primary key</i>
NAME	Nama pengguna
USERNAME	Nama unik pengguna yang digunakan untuk masuk ke dalam sistem
PASSWORD	Kata sandi yang digunakan untuk masuk ke dalam sistem
MAJOR_ID	<i>Foreign key</i> yang merujuk ke tabel <i>Majors</i>
PERSON_ID	ID yang dapat merujuk ke tabel <i>Students</i> atau <i>Lecturers</i>

PERSON_TYPE	Keterangan yang menyatakan apakah ID dalam kolom PERSON_ID berasal dari tabel <i>Students</i> atau <i>Lecturers</i>
ROLE_ID	ID hak akses pengguna yang sedang aktif. Kolom ini tidak memiliki <i>foreign key</i> karena terdapat kondisi di mana kolom bernilai null, yaitu saat inisialisasi
REMEMBER_TOKEN	Token yang digunakan sebagai pengenalan sesi login pengguna
CREATED_AT	Tanggal pertama kali disimpan
UPDATED_AT	Tanggal terakhir diubah

Tabel 3.53 menunjukkan rancangan tabel *permission role*. Tabel tersebut merupakan tabel *pivot* antara tabel *permissions* dan *roles* yang menyimpan rincian izin akses ke halaman-halaman tertentu yang dapat diakses oleh peran atau hak akses tertentu.

Tabel 3.53 Rancangan Tabel Permission Role

Nama Kolom	Keterangan
PERMISSION_ID	<i>Primary key</i> sekaligus <i>foreign key</i> yang merujuk ke tabel <i>Permissions</i>
ROLE_ID	<i>Primary key</i> sekaligus <i>foreign key</i> yang merujuk ke tabel <i>Roles</i>
WORKFLOW_ID	<i>Primary key</i> sekaligus <i>foreign key</i> yang merujuk ke tabel <i>Workflows</i>

Tabel 3.54 menunjukkan rancangan tabel *role user*. Tabel tersebut merupakan tabel *pivot* antara tabel *roles* dan *users* yang menyimpan rincian peran hak akses yang dimiliki oleh pengguna.

Tabel 3.54 Rancangan Tabel Role User

Nama Kolom	Keterangan
ROLE_ID	<i>Primary key</i> sekaligus <i>foreign key</i> yang merujuk ke tabel <i>Roles</i>
USER_ID	<i>Primary key</i> sekaligus <i>foreign key</i> yang merujuk ke tabel <i>Users</i>

Tabel 3.55 menunjukkan rancangan tabel *workflows*. Tabel tersebut menyimpan data paket aliran kerja.

Tabel 3.55 Rancangan Tabel *Workflows*

Nama Kolom	Keterangan
ID	<i>Primary key</i>
NAME	Nama <i>workflow</i>
DESCRIPTION	Keterangan penjas mengenai <i>workflow</i>

Tabel 3.56 menunjukkan rancangan tabel *menus*. Tabel tersebut menyimpan data menu-menu yang bisa diakses oleh pengguna.

Tabel 3.56 Rancangan Tabel *Menus*

Nama Kolom	Keterangan
ID	<i>Primary key</i>
NAME	Nama <i>menu</i>
URL	Alamat rujukan menu dalam sistem
ICON	Keterangan mengenai gambar menu

Tabel 3.57 menunjukkan rancangan tabel *menu role*. Tabel tersebut merupakan tabel *pivot* antara tabel *menus* dan *roles* yang menyimpan rincian menu-menu yang dapat diakses oleh peran tertentu.

Tabel 3.57 Rancangan Tabel *Menu Role*

Nama Kolom	Keterangan
MENU_ID	<i>Primary key</i> sekaligus <i>foreign key</i> yang merujuk ke tabel <i>Menus</i>
ROLE_ID	<i>Primary key</i> sekaligus <i>foreign key</i> yang merujuk ke tabel <i>Roles</i>
WORKFLOW_ID	<i>Primary key</i> sekaligus <i>foreign key</i> yang merujuk ke tabel <i>Workflows</i>

Tabel 3.58 menunjukkan rancangan tabel *parameters*. Tabel tersebut menyimpan daftar parameter yang bisa disesuaikan untuk setiap jurusan.

Tabel 3.58 Rancangan Tabel *Parameters*

Nama Kolom	Keterangan
ID	<i>Primary key</i>
CODE	Kode yang dirujuk oleh kode program
NAME	Nama <i>parameter</i>

DESCRIPTION	Keterangan mengenai parameter
-------------	-------------------------------

Tabel 3.59 menunjukkan rancangan tabel *major parameter*. Tabel tersebut merupakan tabel *pivot* antara tabel *majors* dan *parameters* yang menyimpan nilai-nilai paramter yang dimiliki oleh jurusan.

Tabel 3.59 Rancangan Tabel Major Parameter

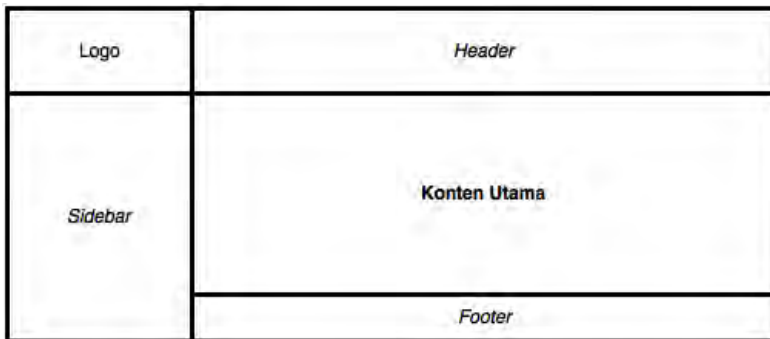
Nama Kolom	Keterangan
MAJOR_ID	<i>Primary key</i> sekaligus <i>foreign key</i> yang merujuk ke tabel <i>Majors</i>
PARAMETER_ID	<i>Primary key</i> sekaligus <i>foreign key</i> yang merujuk ke tabel <i>Parameters</i>
VALUE	Nilai parameter

3.2.4 Perancangan Antarmuka Pengguna

Pada subbab ini akan dijelaskan mengenai rancangan antarmuka yang akan menghubungkan pengguna dengan sistem. Rancangan yang dibahas meliputi kontrol pada antarmuka dan ketentuan input.

3.2.4.1 Antarmuka Kerangka Tata Letak

Kerangka tata letak digunakan cetak biru bagi antarmuka lainnya. Penggunaan kerangka tata letak dimaksudkan agar tidak jadi perulangan dalam membuat antarmuka yang identik. Dalam kerangka tata letak tersebut, terdapat elemen-elemen antarmuka pengguna yang bersifat statis, yaitu bagian atas (*header*), bagian samping (*sidebar*), dan bagian bawah tampilan (*footer*). Rancangan antarmuka kerangka tata letak ditunjukkan pada Gambar 3.41.



Gambar 3.41 Antarmuka Kerangka Tata Letak

3.2.4.2 Antarmuka Halaman Login

Halaman ini merupakan tampilan yang muncul ketika pengguna hendak masuk ke dalam sistem. Terdapat dua isian dan sebuah tombol dalam halaman ini. Isian-isian tersebut adalah nama pengguna dan kata sandi. Isian nama pengguna bertipe teks, sedangkan isian kata sandi bertipe teks *password*. Rancangan antarmuka halaman login ditunjukkan pada Gambar 3.42.

Nama Pengguna

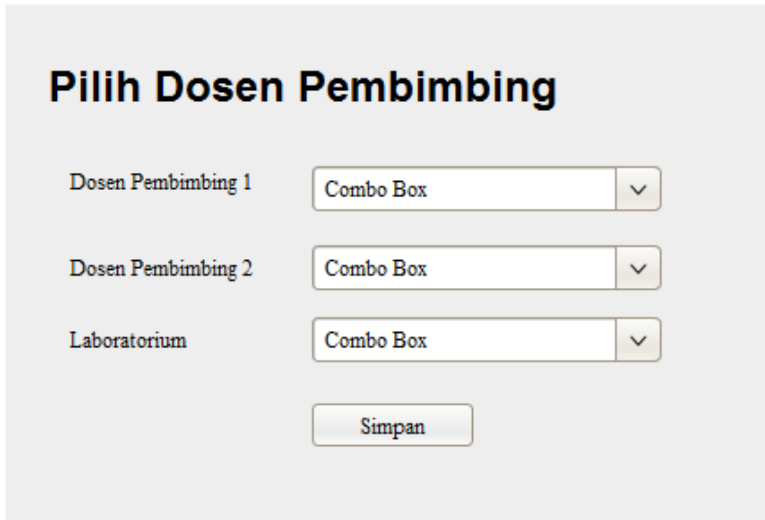
Kata Sandi

Login

Gambar 3.42 Rancangan Antarmuka Halaman Login

3.2.4.3 Antarmuka Halaman Pemilihan Dosen Pembimbing

Halaman ini ditampilkan saat mahasiswa memilih dosen pembimbing tugas akhir. Terdapat tiga isian berupa *combo box* dan sebuah tombol simpan. Rancangan antarmuka halaman pemilihan dosen pembimbing ditunjukkan pada Gambar 3.43.



The image shows a web form titled "Pilih Dosen Pembimbing". It contains three rows, each with a label and a dropdown menu. The labels are "Dosen Pembimbing 1", "Dosen Pembimbing 2", and "Laboratorium". Each dropdown menu currently displays "Combo Box" and has a downward arrow icon. Below these three rows is a button labeled "Simpan".

Label	Input Type	Current Value
Dosen Pembimbing 1	Combo Box	Combo Box
Dosen Pembimbing 2	Combo Box	Combo Box
Laboratorium	Combo Box	Combo Box

Simpan

Gambar 3.43 Rancangan Antarmuka Pemilihan Dosen Pembimbing

3.2.4.4 Rancangan Antarmuka Halaman Unggah Proposal

Rancangan antarmuka halaman unggah proposal ditunjukkan pada Gambar 3.44.



Unggah Proposal

Proposal No file chosen

Judul

Abstraksi

Gambar 3.44 Rancangan Antarmuka Halaman Unggah Proposal

3.2.4.5 Rancangan Antarmuka Halaman Pengelolaan Progres

Rancangan antarmuka halaman pengelolaan progres ditunjukkan pada Gambar 3.45.

Kelola Progres

Tambah Progres Bimbingan

Tambah Progres Buku

Progres Bimbingan

No.	Tanggal	Aktivitas	Aksi

Progres Buku

No.	Tanggal	Bab	Persentase	Keterangan	Aksi

Gambar 3.45 Rancangan Antarmuka Halaman Pengelolaan Progres

3.2.4.6 Rancangan Antarmuka Halaman Penambahan Progres Bimbingan

Rancangan antarmuka halaman penambahan progres bimbingan ditunjukkan pada Gambar 3.46.



Tambah Progres Bimbingan

Tanggal

Aktivitas

Simpan

Gambar 3.46 Rancangan Antarmuka Halaman Penambahan Progres Bimbingan

3.2.4.7 Rancangan Antarmuka Halaman Penambahan Progres Buku

Rancangan antarmuka halaman penambahan progres buku ditunjukkan pada Gambar 3.47.

Tambah Progres Buku

Tanggal

Bab

Pencapaian

Progres Buku

Penjelasan

Gambar 3.47 Rancangan Antarmuka Halaman Penambahan Progres Buku

3.2.4.8 Rancangan Antarmuka Halaman Jadwal Sidang Pribadi Mahasiswa

Rancangan antarmuka halaman jadwal sidang pribadi mahasiswa ditunjukkan pada Gambar 3.48.

Jadwal Sidang

No.	Jenis Sidang	Tanggal	Mulai	Selesai	Aksi

Gambar 3.48 Rancangan Antarmuka Halaman Jadwal Sidang Pribadi Mahasiswa

3.2.4.9 Rancangan Antarmuka Halaman Verifikasi Progres

Rancangan antarmuka halaman verifikasi progres ditunjukkan pada Gambar 3.49.

Verifikasi Progres Bimbingan

No.	NRP	Nama	Tanggal	Kegiatan Progres	Aksi
					<input type="button" value="Verifikasi"/> <input type="button" value="Tidak Valid"/>

Gambar 3.49 Rancangan Antarmuka Halaman Verifikasi Progres

3.2.4.10 Rancangan Antarmuka Halaman Jadwal Sidang Pribadi Dosen

Rancangan antarmuka halaman jadwal sidang pribadi dosen ditunjukkan pada Gambar 3.50.

Jadwal Sidang

No.	NRP	Nama	Jenis Sidang	Tanggal	Mulai	Selesai	Ruangan

Gambar 3.50 Rancangan Antarmuka Halaman Jadwal Sidang Pribadi Dosen

3.2.4.11 Rancangan Antarmuka Halaman Kelola Sidang

Rancangan antarmuka halaman kelola sidang ditunjukkan pada Gambar 3.51.

Kelola Sidang

[Tambah Sidang](#)

No.	Nama Mahasiswa	Jenis Sidang	Tanggal	Mulai	Selesai	Ruangan	Aksi

Gambar 3.51 Rancangan Antarmuka Halaman Kelola Sidang

3.2.4.12 Rancangan Antarmuka Halaman Penambahan Sidang

Rancangan antarmuka halaman penambahan sidang ditunjukkan pada Gambar 3.52.

Tambah Sidang Baru

NRP Mahasiswa

Tanggal Sidang

Sesi Sidang ▼

Ruang Sidang ▼

Jenis Sidang ▼

Gambar 3.52 Rancangan Antarmuka Halaman Penambahan Sidang

3.2.4.13 Rancangan Antarmuka Halaman Perubahan Sidang

Rancangan antarmuka halaman perubahan sidang ditunjukkan pada Gambar 3.53.

Ubah Sidang

NRP Mahasiswa

Text box

Tanggal Sidang

Text box

Sesi Sidang

Combo Box

▼

Ruang Sidang

Combo Box

▼

Jenis Sidang

Combo Box

▼

Simpan

Gambar 3.53 Rancangan Antarmuka Halaman Pengubahan Sidang

3.2.4.14 Rancangan Antarmuka Halaman Kelola Data Mahasiswa

Rancangan antarmuka halaman kelola data mahasiswa ditunjukkan pada Gambar 3.54.

Kelola Mahasiswa


Tambah Mahasiswa

No.	Nama	NRP	Jenjang	Jenis Jenjang	Aksi

Gambar 3.54 Rancangan Antarmuka Halaman Kelola Data Mahasiswa

3.2.4.15 Rancangan Antarmuka Halaman Penambahan Mahasiswa

Rancangan antarmuka halaman penambahan mahasiswa ditunjukkan pada Gambar 3.55.



The image shows a web form titled "Tambah Mahasiswa" (Add Student) in a large, bold, black font. Below the title, there are four input fields arranged vertically. The first field is labeled "NRP" and contains the placeholder text "Text box". The second field is labeled "Nama" and also contains the placeholder text "Text box". The third field is labeled "Jenjang Pendidikan" (Education Level) and contains the placeholder text "Combo Box" with a downward arrow icon on the right. The fourth field is labeled "Jenis Jenjang" (Type of Level) and also contains the placeholder text "Combo Box" with a downward arrow icon on the right. At the bottom of the form, there is a button labeled "Simpan" (Save) in a light gray box with a thin border.

Gambar 3.55 Rancangan Antarmuka Halaman Penambahan Mahasiswa

3.2.4.16 Rancangan Antarmuka Halaman Perubahan Mahasiswa

Rancangan antarmuka halaman perubahan mahasiswa ditunjukkan pada Gambar 3.56.

Ubah Mahasiswa

NRP

Nama

Jenjang Pendidikan

▼

Jenis Jenjang

▼

Gambar 3.56 Rancangan Antarmuka Halaman Perubahan Mahasiswa

3.2.4.17 Rancangan Antarmuka Halaman Pengelolaan Nilai

Rancangan antarmuka halaman pengelolaan nilai ditunjukkan pada Gambar 3.57.

Kelola Nilai						
No.	Nama	NRP	Nilai Proposal	Nilai Akhir	Rata-Rata	Aksi

Gambar 3.57 Rancangan Antarmuka Halaman Pengelolaan Nilai

3.2.4.18 Rancangan Antarmuka Halaman Pemasukan Nilai

Rancangan antarmuka halaman pemasukan nilai ditunjukkan pada Gambar 3.58.



The image shows a web form titled "Input Nilai" in a large, bold, black font. Below the title, there are four rows of input fields. The first row has the label "NRP" and the value "123456789". The second row has the label "Nama" and the value "Mahasiswa Teladan". The third row has the label "Nilai Proposal" and a text input field containing "100". The fourth row has the label "Nilai Akhir" and a text input field containing "100". At the bottom of the form is a button labeled "Simpan".

NRP	123456789
Nama	Mahasiswa Teladan
Nilai Proposal	<input type="text" value="100"/>
Nilai Akhir	<input type="text" value="100"/>

Gambar 3.58 Rancangan Antarmuka Halaman Pemasukan Nilai

[Halaman ini sengaja dikosongkan]

BAB 4

IMPLEMENTASI

Pada bab ini akan dibahas mengenai implementasi dari perancangan perangkat lunak. Di dalamnya mencakup proses penerapan dan pengimplementasian data, proses, dan antarmuka yang mengacu pada rancangan yang telah dibahas sebelumnya. Bahasa yang digunakan adalah bahasa pemrograman PHP dengan kerangka kerja pemrograman Laravel.

4.1 Lingkungan Implementasi

Dalam merancang perangkat lunak ini digunakan beberapa perangkat pendukung yang terdiri dari perangkat keras dan perangkat lunak yang digunakan.

4.1.1 Lingkungan Implementasi Perangkat Keras

Perangkat keras yang digunakan dalam pengembangan sistem adalah laptop. Spesifikasi dari perangkat tersebut adalah laptop Lenovo ThinkPad T440, prosesor Intel Core i5-4200U @ 1,60 GHz 2,30 GHz, dan RAM 8 GB.

4.1.2 Lingkungan Implementasi Perangkat Lunak

Spesifikasi perangkat lunak yang digunakan dalam pengembangan sistem adalah sebagai berikut.

- Microsoft Windows 10 Pro 64 bit sebagai sistem operasi.
- Sublime Text 3 sebagai penyunting teks.
- MySQL sebagai server basis data.
- Visual Paradigm 12.2 CE untuk merancang basis data.
- Evolus Pencil untuk merancang antarmuka.
- GitHub sebagai layanan penyedia repositori kode program.

4.2 Implementasi Proses Aplikasi

Pada bagian ini akan dijelaskan mengenai implementasi proses aplikasi ke dalam sebuah kode. Implementasi proses aplikasi ini dilakukan dengan menggunakan bahasa pemrograman PHP dengan kerangka kerja Laravel.

4.2.1 Implementasi Proses Sistem Generik

Proses-proses dalam sistem generik dilakukan oleh beberapa jenis pengguna, yaitu mahasiswa, dosen, koordinator, dan administrator. Penjelasan mengenai setiap proses akan dibahas dalam subbab tersendiri.

4.2.1.1 Implementasi Proses Pemilihan Dosen Pembimbing

Proses ini dilakukan oleh mahasiswa sebagai langkah awal menempuh prosedur tugas akhir. Dalam proses ini mahasiswa memilih dosen pembimbing dan laboratorium tugas akhir. Jumlah dosen pembimbing yang dipilih berbeda-beda tergantung dari nilai parameter jumlah dosen pembimbing per mahasiswa yang telah diatur. Pengaturan parameter tersebut merupakan hak dari koordinator.

Permintaan HTTP untuk proses ini dilayani oleh pengendali *ProjectController*. Sedangkan, logika utama proses dieksekusi oleh lapisan layanan (*service*) *ProjectService*. Kode program proses pemilihan dosen pembimbing dapat dilihat pada Kode Sumber 4.1.

```
public function postChooseLecturer(Request $request,
ProjectService $service){
    $lecturer_ids = $request->input('lecturer_ids');
    $lecturers     = Lecturer::whereIn('id', $lecturer_ids)->get();
    $interestArea = InterestArea::find($request-
>input('interest_area_id'));
    $student      = Auth::user()->person;
    $service->chooseLecturer($lecturers, $student, $interestArea);
    flash()->success('Data pembimbing berhasil disimpan.');
```

```
return redirect()->back();
}
```

Kode Sumber 4.1 Pemilihan Dosen Pembimbing

4.2.1.2 Implementasi Proses Pengunggahan Proposal

Proses pengunggahan proposal dilakukan setelah mahasiswa memilih dosen pembimbing. Proses ini pada dasarnya hanya merupakan proses penyimpanan berkas proposal ke dalam sistem. Mahasiswa berhak mengunggah proposal lebih dari satu kali apabila terdapat revisi yang harus diunggah kembali. Kode program pengunggahan proposal dapat dilihat pada Kode Sumber 4.2.

```

public function postUploadProposal(Request $request,
ProjectService $service)
{
    $student          = Auth::user()->person;
    $proposal          = new Proposal;
    $proposal->title    = $request->input('title');
    $proposal->file      = $request->file('file');
    $proposal->abstract  = $request->input('abstract');
    $service->uploadProposal($student, $proposal);
    flash()->success('Proposal berhasil diunggah.');
```

```

    return redirect()->back();
}

```

Kode Sumber 4.2 Pengunggahan Proposal

4.2.1.3 Implementasi Proses Pengisian Progres Bimbingan

Pengisian progres bimbingan dilakukan mahasiswa untuk melaporkan kegiatan bimbingan yang dilakukan bersama dosen pembimbing. Terdapat dua jenis progres, yaitu progres bimbingan dan progres buku. Progres bimbingan merupakan progres bimbingan yang bersifat umum terkait tugas akhir. Sedangkan, progres buku merupakan progres khusus terkait pengerjaan buku tugas akhir.

Proses ini dikelola oleh pengendali *ProgressController*. Kedua jenis progres tersebut dieksekusi oleh sebuah pengendali karena keduanya memiliki konteks yang sama dan untuk menyederhanakan implementasi. Kelas pengendali *ProgressController* menggunakan sebuah layanan yaitu *FileService*. Kelas *FileService* digunakan untuk mengunggah berkas tugas progres buku. Implementasi kode program pengisian progres bimbingan dapat dilihat pada Kode Sumber 4.3.

```

public function postCreateBookProgress(Request $request,
FileService $file) {
    $student                = Auth::user()->person;
    $bookProgress            = new BookProgress;
    $bookProgress->event      = Carbon::parse($request-
>input('event'));
    $bookProgress->performance = $request->input('performance');
    $bookProgress->explanation = $request->input('explanation');
    if ($request->hasFile('file')) {
        $destinationPath    = 'upload/book-progress/' . $student-
>id . '/' . str_random(8) . '/';
        $bookProgress->file = $file->upload($request-
>file('file'), $destinationPath);
    }
    $chapter = Chapter::find($request->input('chapter_id'));
    $bookProgress->chapter()->associate($chapter);
    $bookProgress->student()->associate($student);
    $bookProgress->project()->associate($student->project);
    $status = ProgressStatus::where('code', 'belum diverifikasi')-
>first();
    $bookProgress->status()->associate($status);
    $bookProgress->major()->associate($student->major);
    $bookProgress->save();
    flash()->success('Progress berhasil disimpan.');
```

```

    return redirect('final-project/progress');
```

Kode Sumber 4.3 Progres Bimbingan

4.2.1.4 Implementasi Proses Verifikasi Progres Bimbingan

Verifikasi progres bimbingan dilakukan oleh dosen pembimbing. Apabila terdapat lebih dari satu dosen pembimbing, masing-masing dosen berhak melakukan verifikasi. Verifikasi meliputi menerima progres sebagai kegiatan yang valid atau menolaknya. Proses ini dieksekusi oleh pengendali *ProgresController*. Kode program proses verifikasi progres bimbingan dapat dilihat pada Kode Sumber 4.4.

```

public function verification()
{
    $lecturer      = Auth::user()->person;
    $allProgresses = collect();
    foreach ($lecturer->projects as $project) {
        $allProgresses = $allProgresses->merge($project-
>progresses);
        $allProgresses = $allProgresses->merge($project-
>bookProgresses);
    }

    $statusBelumDiverifikasi = ProgressStatus::where('code',
'belum diverifikasi')->firstOrFail();

    return view('pages.its.lecturer.progress-verification',
compact('allProgresses', 'statusBelumDiverifikasi'));
}

```

Kode Sumber 4.4 Verifikasi Progres Bimbingan

4.2.1.5 Implementasi Proses Pengelolaan Jadwal Sidang

Pengelolaan jadwal sidang dilakukan oleh koordinator. Pembuatan jadwal sidang dilakukan dalam sebuah halaman dengan mengisi isian NRP mahasiswa, tanggal sidang, sesi sidang, ruang sidang, jenis sidang, dan dosen penguji. NRP mahasiswa digunakan untuk mencari entitas tugas akhir mahasiswa yang direpresentasikan dalam kelas *Project*. Sedangkan, entitas penyimpan jadwal sidang itu sendiri adalah model kelas *Meeting*.

Proses pengelolaan jadwal sidang dieksekusi oleh pengendali *MeetingController*. Dalam proses pembuatan jadwal sidang, pengendali tersebut memanfaatkan layanan *ProjectService* untuk mendapatkan data tugas akhir mahasiswa. Kode program proses pengelolaan jadwal sidang dapat dilihat pada Kode Sumber 4.5.

```

public function lecturerSchedule()
{
    $lecturer = Auth::user()->person;
    $project_ids = $lecturer->projects->pluck('id');

    $meetings = Meeting::with('student', 'room', 'type',
'session')->latest()
        ->whereIn('project_id', $project_ids)->get();
    return view('pages.its.lecturer.meeting',
compact('meetings'));
}

public function studentSchedule()
{
    $student = Auth::user()->person;

    $meetings = Meeting::with('student', 'room', 'type',
'session')->latest()
        ->where('student_id', $student->id)->get();
    return view('pages.its.student.meeting', compact('meetings'));
}

```

Kode Sumber 4.5 Jadwal Sidang

4.2.1.6 Implementasi Proses Pengelolaan Nilai

Proses pengelolaan nilai dilakukan oleh administrator. Administrator dapat merupakan petugas jurusan yang bertugas membantu tugas koordinator tugas akhir. Administrator bertugas memasukkan nilai tugas akhir mahasiswa ke dalam sistem setelah nilai muncul berdasarkan hasil sidang. Penyerahan nilai dari dosen penguji kepada administrator dilakukan di luar sistem.

Proses pengelolaan nilai ditangani oleh pengendali *GradeController*. Kode program pengelolaan nilai dapat dilihat pada Kode Sumber 4.6.


```

public function edit($id)
{
    $grade = Project::find($id);
    $title = 'Ubah Nilai';
    return view('pages.grade.create', compact('grade', 'title'));
}

public function update(Request $request, $id)
{
    DB::transaction(function () use ($request, $id) {
        $grade = Project::find($id);
        $grade->update($request->only('proposal_grade',
'final_grade'));
        $grade->average_grade = ($grade->proposal_grade + $grade-
>final_grade) / 2;
        $grade->save();
        flash()->success('Data jurusan berhasil diperbarui.');
```

Kode Sumber 4.6 Pengelolaan Nilai

4.2.2 Implementasi Proses Pengaturan Aliran Kerja

Proses pengaturan aliran kerja merupakan proses pengelolaan paket menu dan hak akses jurusan. Proses ini dilakukan oleh administrator pusat ITS. Pengelolaan paket menu dan hak akses membutuhkan wawasan mengenai menu dan fitur sistem sehingga hak akses pengaturan aliran kerja ini diserahkan kepada pihak LPTSI. Proses-proses yang rinci mengenai pengaturan aliran kerja dijelaskan pada subbab tersendiri dalam bab ini.

4.2.2.1 Implementasi Proses Pengelolaan Jurusan

Proses pengelolaan jurusan merupakan proses manajemen data jurusan yang ada dalam sistem. Dalam proses ini, jurusan baru dapat ditambahkan. Jurusan direpresentasikan oleh model *Major*. Eksekusi proses pengelolaan jurusan dilakukan oleh pengendali *MajorController*. Proses pengelolaan jurusan pada dasarnya merupakan proses CRUD (*Create, Read, Update, Delete*) biasa. Kode program proses pengelolaan jurusan dapat dilihat pada Kode Sumber 4.7.

```

public function create()
{
    $major      = new Major;
    $title      = 'Tambah Jurusan';
    $workflows  = Workflow::get()->sortBy('name')->pluck('name',
'id');
    return view('pages.major.create', compact('major', 'title',
'workflows'));
}

public function store(Request $request)
{
    $major      = Major::create($request->only('name', 'code'));
    $workflow   = Workflow::find($request->input('workflow_id'));
    $major->workflow()->associate($workflow)->save();
    flash()->success('Jurusan baru berhasil ditambahkan.');
```

```

    return redirect('major');
}

```

Kode Sumber 4.7 Pengelolaan Jurusan

4.2.2.2 Implementasi Proses Pengelolaan Menu

Menu merupakan elemen halaman yang memiliki fungsi navigasi sistem. Menu dibuat untuk memberikan informasi mengenai fitur apa saja yang dapat diakses oleh pengguna. Pengelolaan menu dapat dilakukan oleh seorang administrator sistem. Dalam proses pengelolaan menu ini, menu baru dapat ditambahkan. Menu direpresentasikan oleh model *Menu*. Proses pengelolaan menu pada dasarnya merupakan proses CRUD (Create, Read, Update, Delete) biasa. Kode program proses pengelolaan menu dapat dilihat pada Kode Sumber 4.8.

```

public function create()
{
    $menu          = new Menu;
    $title         = 'Tambah Menu';
    $permissions   = Permission::get()->sortBy('name');
    return view('pages.menu.create', compact('menu', 'title',
    'permissions'));
}

public function store(Request $request)
{
    $menu = Menu::create($request->only('name', 'url', 'icon'));
    $permission_ids = $request->input('permission_ids');
    $menu->permissions()->sync($permission_ids);
    flash()->success('Menu baru berhasil ditambahkan. ');
    return redirect('menu');
}

```

Kode Sumber 4.8 Pengelolaan Menu

4.2.2.3 Implementasi Proses Pengelolaan Hak Akses

Hak akses adalah kewenangan untuk menjalankan fitur tertentu dalam sistem. Hak akses terkait erat dengan alamat atau rute yang bisa diakses dalam sistem. Setiap fitur yang dibatasi jenis penggunaannya harus diatur hak aksesnya dalam fitur pengelolaan hak akses.

Pengelolaan hak akses dilakukan oleh administrator pusat. Proses pengelolaan hak akses pada dasarnya merupakan proses CRUD (Create, Read, Update, Delete) biasa. Kode program proses pengelolaan hak akses dapat dilihat pada Kode Sumber 4.9.

```

public function create()
{
    $permission = new Permission;
    $title      = 'Tambah Permission';
    return view('pages.permission.create', compact('permission',
'title'));
}

public function store(Request $request)
{
    Permission::create($request->only('name', 'display_name',
'description'));
    flash()->success('Permission baru berhasil ditambahkan.');
```

```

    return redirect('permission');
}

```

Kode Sumber 4.9 Pemilihan Pengelolaan Hak Akses

4.2.2.4 Implementasi Proses Pengelolaan Peran

Peran atau jenis pengguna adalah entitas yang digunakan untuk mengategorikan pengguna. Peran yang direpresentasikan oleh model *Role* memiliki relasi utama dengan menu dan hak akses. Peran berguna untuk mengategorikan pengguna berdasarkan menu dan hak aksesnya.

Pengelolaan peran dilakukan oleh administrator pusat. Proses pengelolaan peran pada dasarnya merupakan proses CRUD (Create, Read, Update, Delete) biasa. Kode program proses pengelolaan peran dapat dilihat pada Kode Sumber 4.10.

```

public function create()
{
    $role = new Role;
    $title = 'Tambah Role';
    return view('pages.role.create', compact('role', 'title'));
}

public function store(Request $request)
{
    Role::create($request->only('name', 'display_name',
'description'));
    flash()->success('Role baru berhasil ditambahkan.');
```

```

    return redirect('role');
}

```

Kode Sumber 4.10 Pemilihan Pengelolaan Peran

4.2.2.5 Implementasi Proses Pengelolaan Aliran Kerja

Dalam proses ini, hal pertama yang harus dilakukan adalah menambahkan aliran kerja. Aliran kerja tersebut akan disimpan sebagai model *Workflow* dalam tabel *workflows*. Setelah itu, administrator pusat harus memilih menu dan hak akses dari daftar yang tersedia untuk ditambahkan sebagai menu apa saja yang dapat diakses dalam aliran kerja tersebut. Pemilihan menu dan hak akses disertai dengan aksi pemetaan kedua entitas tersebut kepada peran atau jenis pengguna yang dapat mengaksesnya. Data relasi tersebut kemudian akan disimpan oleh sistem dalam dua tabel, yaitu tabel *menu_role* dan *permission_role*. Atribut *workflow_id* pada kedua tabel tersebut akan diisi sesuai dengan nilai kunci utama (*primary key*) dari aliran kerja (*workflow*) yang baru ditambahkan. Keseluruhan informasi dapat disimpan ketika semua menu dan hak akses telah terpetakan terhadap peran penggunaanya.

Proses pengelolaan aliran kerja ditangani oleh pengendali *WorkflowController*. Kode program proses pengelolaan aliran kerja tersebut dapat dilihat pada Kode Sumber 4.11.

```
public function mapping($id, Request $request)
{
    $workflow = Workflow::find($id);
    if ($request->isMethod('get')) {
        $workflow->load('menus');
        $title = 'Pemetaan Proses Aliran Kerja';
        $roles = Role::get()->pluck('display_name', 'id');
        $menus = Menu::get()->pluck('name', 'id');
        return view('pages.workflow.mapping', compact('workflow',
        'title', 'roles', 'menus'));
    } else {
        $workflow->menus()->detach();
        $workflow->permissions()->detach();
        $role_ids = $request->input('role_ids');
        $menu_ids = $request->input('menu_ids');
        DB::transaction(function () use ($role_ids, $workflow,
        $menu_ids) {
            foreach ($role_ids as $key => $role_id) {
                $workflow->menus()->attach($menu_ids[$key],
                ['role_id' => $role_id]);
                $menu = Menu::find($menu_ids[$key]);
```

```

        foreach ($menu->permissions as $permission) {
            $workflow->permissions()->attach($permission-
>id, ['role_id' => $role_id]);
        }

        flash()->success('Pemetaan proses aliran kerja
berhasil disimpan.');
```

Kode Sumber 4.11 Pengelolaan Aliran Kerja

4.2.3 Implementasi Proses Pengaturan Parameter

Proses pengaturan parameter dilakukan oleh koordinator jurusan. Pengaturan parameter bertujuan untuk menentukan perilaku aliran kerja sistem.

Parameter direpresentasikan oleh model *Parameters*. Model tersebut mewakili informasi parameter yang meliputi nama, kode, dan deskripsi parameter. Data master parameter dimasukkan ke dalam sistem melalui skrip *seeder*.

Nilai parameter yang dimasukkan oleh pengguna disimpan dalam tabel *major_parameter*. Tabel tersebut pada dasarnya merupakan tabel perantara (*pivot*) pada relasi *many-to-many* antara tabel *majors* dan tabel *parameters*.

Proses pengaturan parameter ditangani oleh kode pengelola *ParameterController*. Kode program terkait proses pengaturan parameter dapat dilihat pada Kode Sumber 4.12.

```

public function majorParameter(Request $request)
{
    $major = auth()->user()->major;
    if ($request->isMethod('get')) {
        $parameters = Parameter::get();
        $major->load('parameters');
        $majorParameters = $major->parameters-
>pluck('pivot.value', 'id');
        $title          = 'Pengaturan Parameter';
```

```

        return view('pages.parameter.major-parameter',
compact('majorParameters', 'parameters', 'major', 'title'));
    } else {
        $major->parameters()->detach();
        $parameter_ids = $request->input('parameter_ids');
        $values         = $request->input('values');
        foreach ($parameter_ids as $key => $parameter_id) {
            $major->parameters()->attach($parameter_id, ['value'
=> $values[$key]]);
        }

        flash()->success('Pengaturan parameter berhasil
disimpan.');
```

```

        return redirect()->back();
    }
}

```

Kode Sumber 4.12 Pengaturan Parameter

4.3 Implementasi Antarmuka Pengguna

Pada bagian ini akan dibahas implementasi antarmuka berdasarkan rancangan antarmuka yang telah dibahas pada bab 3. Lapisan antarmuka merupakan lapisan yang bertugas mengatur tampilan sistem agar dapat berinteraksi dengan pengguna. Lapisan antarmuka berinteraksi langsung dengan lapisan pengendali (*controller*).

Kode program antarmuka terletak pada direktori *app/resources/views*. Di dalam direktori tersebut, berkas-berkas kode program telah dikategorikan berdasarkan kegunaannya. Terdapat enam buah subdirektori pada direktori *views* tersebut, yaitu *errors*, *layouts*, *pages*, *partials*, *scripts*, dan *vendors*. Direktori *errors* memuat berkas-berkas antarmuka yang dapat digunakan untuk menampilkan pesan kesalahan; direktori *layouts* memuat cetak biru tata letak antarmuka yang digunakan oleh setiap halaman; direktori *pages* merupakan lokasi berkas antarmuka yang mewakili tiap-tiap halaman dalam tampilan sistem; direktori *partials* memuat kode-kode yang ditujukan untuk digunakan kembali oleh elemen antarmuka lainnya; direktori *scripts* memuat kode-kode JavaScript; sedangkan, direktori *vendors* memuat kode program antarmuka yang berasal dari pustaka pembantu.

Antarmuka sistem dikembangkan menggunakan beberapa pustaka. Beberapa di antaranya adalah:

- Laravel Blade sebagai skrip bahasa utama;
- AdminLTE sebagai kerangka kerja CSS;
- jQuery sebagai kerangka kerja JavaScript, dan;
- DataTables sebagai pustaka antarmuka tabel.

4.4 Implementasi Basis Data

Pada bab ini akan dijelaskan mengenai implementasi basis data menggunakan Laravel pada basis data MySQL. Implementasi ini meliputi implementasi struktur basis data dan implementasi model.

4.4.1 Implementasi Struktur Basis Data

Implementasi struktur basis data dilakukan dengan memanfaatkan *migration* pada Laravel. Terdapat dua kelas yang memuat informasi skema database, yaitu kelas *InitialSchema* dan *EntrustSetupTables*. Kelas *InitialSchema* memuat tabel-tabel utama, sedangkan kelas *EntrustSetupTables* memuat tabel-tabel untuk keperluan *authentication* dan *authorization*. Implementasi kode program *migration* tersebut dapat dilihat pada Kode Sumber A.1 pada lampiran.

Untuk menjalankan dan menerapkan skema database yang telah dirancang dalam *migration*, sebuah perintah harus dijalankan melalui *terminal*. Perintah tersebut adalah *php artisan migrate*. Perintah tersebut akan mengeksekusi seluruh kode program *migration* yang ada, menerjemahkan kelas ke dalam kode SQL sesuai dengan sintaksis RDBMS yang dipilih, hingga pada akhirnya implementasi stuktur basis data selesai.

4.4.2 Implementasi Model

Model merupakan representasi dari entitas-entitas utama dalam basis data. Model memuat informasi mengenai kolom-kolom (atribut) dalam tabel dan relasi tabel tersebut dengan tabel lainnya. Beberapa implementasi model dalam sistem juga memuat fungsi-fungsi khusus.

Seluruh model diimplementasikan di bawah *namespace Simta\Models*. Setiap model merupakan turunan dari kelas *Illuminate\Database\Eloquent\Model*. Kelas model tersebut merupakan sebuah *active record*, sehingga setiap kelas turunannya dapat memanfaatkan sifat tersebut untuk melakukan kueri dengan sederhana. Di dalam kode program, *active record* dimanfaatkan dalam beberapa lokasi seperti *controller* dan *service*.

Tidak semua tabel memiliki model yang merepresentasikannya. Tabel-tabel *pivot* yang merupakan tabel perantara dalam relasi *many-to-many* tidak memiliki model. Kode program seluruh model dapat dilihat pada lampiran, yaitu Kode Sumber A.2 hingga Kode Sumber A.24.

[Halaman ini sengaja dikosongkan]

BAB 5

PENGUJIAN DAN EVALUASI

Bab ini membahas pengujian dan evaluasi pada aplikasi yang dikembangkan. Pengujian yang dilakukan adalah pengujian terhadap kebutuhan fungsionalitas sistem. Pengujian fungsionalitas mengacu pada kasus penggunaan pada bab tiga. Hasil evaluasi menjabarkan tentang rangkuman hasil pengujian pada bagian akhir bab ini.

5.1 Lingkungan Pengujian

Lingkungan pengujian sistem pada pengerjaan tugas akhir ini dilakukan pada lingkungan dan alat kakas sebagai berikut.

- Jenis Perangkat : Laptop
- Tipe Perangkat : Lenovo ThinkPad T440
- Prosesor : Intel Core i5-4200U 1.6 GHz Dual Core
- Penyimpanan : 256 GB
- RAM : 8 GB
- Sistem Operasi : Windows 10 Pro

5.2 Skenario Pengujian

Pada bagian ini akan dijelaskan tentang skenario pengujian yang dilakukan. Pengujian yang dilakukan adalah pengujian kebutuhan fungsional. Pengujian kebutuhan fungsional sistem dilakukan dengan menggunakan metode kotak hitam (*black box*). Metode ini menekankan pada kesesuaian hasil keluaran sistem.

5.2.1 Pengujian Fungsionalitas

Subbab ini menjelaskan tentang skenario pengujian fungsionalitas perangkat lunak pada tugas akhir ini. Pengujian didokumentasikan secara sistematis sebagai tolok ukur keberhasilan sistem. Pengujian dilakukan oleh pengembang. Berikut ini subbab-subbab yang menjabarkan pengujian pada kebutuhan fungsionalitas sistem.

5.2.1.1 Pengujian Fitur Memilih Dosen Pembimbing

Pengujian fitur ini adalah pemilihan dosen pembimbing melalui hak akses mahasiswa. Rincian pengujian dapat dilihat pada Tabel 5.1, sedangkan hasil pengujian dapat dilihat pada Gambar 5.1.

Tabel 5.1 Pengujian Fitur Memilih Dosen Pembimbing

ID	FT-01	
Referensi Kasus Penggunaan	FR-01	
Nama	Pengujian fitur memilih dosen pembimbing	
Tujuan Pengujian	Menguji fitur memilih dosen pembimbing	
Skenario	Akun yang digunakan adalah mahasiswa	
Kondisi Awal	Pengguna telah login dan berada di halaman beranda	
Langkah Pengujian	Respon Sistem yang Diharapkan	Berhasil
Pengguna memilih menu Pilih Dosen Pembimbing	Sistem menampilkan halaman Pemilihan Dosen Pembimbing	√
Pengguna memilih dosen pembimbing dan laboratorium pada isian yang tersedia, kemudian menekan tombol Simpan	Sistem menyimpan dosen pembimbing dan laboratorium yang dipilih dan menampilkan hasil pilihan pengguna yang telah disimpan	√
Hasil Akhir yang Diharapkan	Sistem menampilkan halaman pemilihan dosen pembimbing dan mampu menyimpan data dosen dan laboratorium bagi mahasiswa pilih	
Hasil Akhir yang Didapat	Sistem dapat menyimpan dosen pembimbing dan laboratorium yang dipilih mahasiswa	
Hasil Pengujian	Berhasil	

Data pembimbing berhasil disimpan.

Pilih Dosen Pembimbing

Dosen Pembimbing 1

Umi Laili Yuhana

Dosen Pembimbing 2

Diana Purwitasari

Laboratorium

Rekayasa Perangkat Lunak

Simpan

Gambar 5.1 Hasil Pengujian Fitur Memilih Dosen Pembimbing

5.2.1.2 Pengujian Fitur Mengunggah Berkas Proposal

Pengujian ini adalah pengunggahan proposal beserta pengisian data judul dan abstraksi tugas akhir. Rincian pengujian dapat dilihat pada Tabel 5.2, sedangkan hasil pengujian dapat dilihat pada Gambar 5.2.

Tabel 5.2 Pengujian Fitur Mengunggah Berkas Proposal

ID	FT-02	
Referensi Kasus Penggunaan	FR-02	
Nama	Pengujian fitur mengunggah berkas proposal	
Tujuan Pengujian	Menguji fitur mengunggah berkas proposal	
Skenario	Akun yang digunakan adalah mahasiswa	
Kondisi Awal	Pengguna telah login dan berada di halaman beranda	
Langkah Pengujian	Respon Sistem yang Diharapkan	Berhasil
Pengguna memilih menu Unggah Proposal	Sistem menampilkan halaman Unggah Proposal	√
Pengguna memilih berkas proposal yang hendak diunggah	Sistem menampilkan tombol pilih berkas dan jendela pencarian berkas proposal	√

Pengguna mengisi isian judul dan abstraksi tugas akhir, kemudian menekan tombol Simpan	Sistem menyimpan proposal beserta data yang diisikan pengguna	✓
Hasil Akhir yang Diharapkan	Sistem menampilkan halaman Unggah Proposal dan mampu menangani proses pengunggahan berkas proposal serta menyimpan data judul dan abstraksi tugas akhir	
Hasil Akhir yang Didapat	Sistem dapat menampilkan halaman Unggah Proposal dan proposal yang diunggah dan data yang diisikan berhasil tersimpan	
Hasil Pengujian	Berhasil	

Upload Proposal

#	Proposal	Tanggal Upload	Aksi
1	Konstruksi Sistem Informasi Administrasi Tugas Akhir Berbasis Aliran Kerja Untuk Institut Teknologi Sepuluh Nopember Menggunakan Kerangka Kerja Pemrograman Laravel	11-01-2016	Download
2	Konstruksi Sistem Informasi Administrasi Tugas Akhir Berbasis Aliran Kerja Untuk Institut Teknologi Sepuluh Nopember Menggunakan Kerangka Kerja Pemrograman Laravel	11-01-2016	Download

Proposal No file chosen

Judul Konstruksi Sistem Informasi Administrasi Tugas Akhir Berbasis Aliran Kerja Untuk Institut Teknologi Sepuluh Nopem

Abstraksi Mengelola dan memantau proses pengerjaan tugas akhir mahasiswa merupakan salah satu kewajiban setiap jurusan di Institut Teknologi Sepuluh Nopember (ITS). Setiap jurusan memiliki cara yang berbeda-beda dalam mengelola dan memantau proses pengerjaan tugas akhir mahasiswa tersebut. Dalam pengelolaan dan pemantauan tersebut, ada jurusan yang memanfaatkan sistem informasi terintegrasi dalam jaringan dan ada pula yang tidak, yaitu di mana semua proses masih dilakukan secara manual dan di luar jaringan. Pemanfaatan sistem informasi dalam mengelola dan memantau tugas akhir mahasiswa dapat memberikan keuntungan kepada jurusan pengguna. Namun, usaha setiap jurusan untuk mengembangkan sistem informasinya masing-masing dapat terhambat oleh kendala tersendiri di masa depan, seperti perubahan kebijakan yang mempengaruhi proses pengelolaan dan pemantauan tugas akhir di semua jurusan di ITS. Oleh

Gambar 5.2 Hasil Pengujian Fitur Mengunggah Berkas Proposal

5.2.1.3 Pengujian Fitur Mengelola Progres Bimbingan Dan Progres Buku

Pengujian ini meliputi menambahkan, mengubah, dan menghapus progres bimbingan dan progres buku. Rincian pengujian dapat dilihat pada Tabel 5.3, sedangkan hasil pengujian dapat dilihat pada Gambar 5.3 hingga Gambar 5.8.

Tabel 5.3 Pengujian Fitur Mengelola Progres Bimbingan dan Progres Buku

ID	FT-03	
Referensi Kasus Penggunaan	FR-03	
Nama	Pengujian fitur mengelola progres bimbingan dan progres buku	
Tujuan Pengujian	Menguji fitur mengelola progres bimbingan dan progres buku	
Skenario	Akun yang digunakan adalah mahasiswa	
Kondisi Awal	Pengguna telah login dan berada di halaman beranda	
Langkah Pengujian	Respon Sistem yang Diharapkan	Berhasil
Pengguna memilih menu Progres Bimbingan	Sistem menampilkan halaman Progres Bimbingan dan Progres Buku	√
Pengguna menekan tombol Tambah Progres Bimbingan	Sistem menampilkan halaman penambahan progres bimbingan	√
Pengguna mengisi isian progres bimbingan, kemudian menekan tombol Simpan	Sistem menyimpan progres bimbingan mahasiswa	√
Pengguna menekan tombol Edit pada progres bimbingan yang hendak diubah	Sistem menampilkan halaman perubahan progres bimbingan	√
Pengguna mengisi isian untuk mengubah data, kemudian menekan tombol simpan	Sistem menyimpan perubahan yang dilakukan	√
Pengguna menekan tombol Hapus pada progres bimbingan yang hendak dihapus	Sistem menghapus progres bimbingan yang dipilih	√
Pengguna menekan tombol Tambah Progres Buku	Sistem menampilkan halaman penambahan progres buku	√
Pengguna mengisi isian progres buku,	Sistem menyimpan progres buku mahasiswa	√

kemudian menekan tombol Simpan		
Pengguna menekan tombol Edit pada progres buku yang hendak diubah	Sistem menampilkan halaman perubahan progres buku	√
Pengguna mengisi isian untuk mengubah data, kemudian menekan tombol simpan	Sistem menyimpan perubahan yang dilakukan	√
Pengguna menekan tombol Hapus pada progres buku yang hendak dihapus	Sistem menghapus progres buku yang dipilih	√
Hasil Akhir yang Diharapkan	Sistem menampilkan halaman dan melakukan proses penambahan, perubahan, dan penghapusan progres bimbingan serta progres buku	
Hasil Akhir yang Didapat	Sistem dapat menampilkan halaman dan menangani proses penambahan, perubahan, dan penghapusan progres bimbingan serta progres buku	
Hasil Pengujian	Berhasil	

Tambah Progres Bimbingan

Tanggal

Aktifitas Bimbingan

Gambar 5.3 Hasil Pengujian Fitur Penambahan Progres Bimbingan

Edit Progres

Tanggal

11-01-2016

Aktifitas Bimbingan

Progres buku bab analisis

Save

Gambar 5.4 Hasil Pengujian Fitur Pengubahan Progres Bimbingan

Progres Bimbingan

☐ ☐ ☐

Show: 10 entries

Search:

No.	Tanggal	Aktifitas / Topik Bimbingan	Aksi
1	11-01-2016	Progres buku	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>

Showing 1 to 1 of 1 entries

Previous
1
Next

Gambar 5.5 Hasil Pengujian Fitur Melihat Daftar Progres Bimbingan

Tambah Progres Buku

Tanggal	11-01-2016
Bab	Bab 1
Pencapaian	100
Progres Buku TA	<div>Choose file</div> Contoh Proposal TA.pdf
Penjelasan	<div>Penjelasan bab 1</div>

Save

Gambar 5.6 Hasil Pengujian Fitur Penambahan Progres Buku

Edit Progres Buku

Tanggal	11-01-2016
Bab	Bab 1
Pencapaian	100
Progres Buku TA	<input type="button" value="Choose file"/> No file chosen
Penjelasan	Contoh bab 1

Gambar 5.7 Hasil Pengujian Fitur Pengubahan Progres Buku

Progres Buku

Show 10 entries Search:

No.	Tanggal	Bab	Persentase	Keterangan	Aksi
1	11-01-2016	Bab 1	100	Contoh bab 1	<input type="button" value="Edit"/> <input type="button" value="Delete"/>

Showing 1 to 1 of 1 entries

Previous 1 Next

Gambar 5.8 Hasil Pengujian Fitur Melihat Daftar Progres Buku

5.2.1.4 Pengujian Fitur Melihat Jadwal Sidang Pribadi (Mahasiswa)

Pengujian ini adalah melihat halaman jadwal sidang pribadi dari akun mahasiswa. Rincian pengujian dapat dilihat pada Tabel 5.4, sedangkan hasil pengujian dapat dilihat pada Gambar 5.9.

Tabel 5.4 Pengujian Fitur Melihat Jadwal Sidang Pribadi (Mahasiswa)

ID	FT-04	
Referensi Kasus Penggunaan	FR-04	
Nama	Pengujian fitur melihat jadwal sidang pribadi (mahasiswa)	
Tujuan Pengujian	Menguji fitur melihat jadwal sidang pribadi (mahasiswa)	
Skenario	Akun yang digunakan adalah mahasiswa	
Kondisi Awal	Pengguna telah login dan berada di halaman beranda	
Langkah Pengujian	Respon Sistem yang Diharapkan	Berhasil
Pengguna memilih menu Jadwal Sidang	Sistem menampilkan halaman jadwal sidang mahasiswa	√
Hasil Akhir yang Diharapkan	Sistem menampilkan halaman jadwal sidang mahasiswa	
Hasil Akhir yang Didapat	Sistem dapat menampilkan halaman jadwal sidang mahasiswa	
Hasil Pengujian	Berhasil	

#	Jenis Sidang	Tanggal	Mulai	Selesai	Ruangan
1	Sidang Akhir	11-01-2016	08:00:00	10:00:00	R. Sidang 1

Gambar 5.9 Hasil Pengujian Fitur Melihat Jadwal Sidang Pribadi (Mahasiswa)

5.2.1.5 Pengujian Fitur Melihat Laporan Pemantauan Progres Mahasiswa

Pengujian ini adalah melihat laporan pemantauan progres mahasiswa sebagai melalui akun pengguna dosen. Rincian

pengujian dapat dilihat pada Tabel 5.5, sedangkan hasil pengujian dapat dilihat pada Gambar 5.10.

Tabel 5.5 Pengujian Fitur Melihat Laporan Pemantauan Progres Mahasiswa

ID	FT-05	
Referensi Kasus Penggunaan	FR-05	
Nama	Pengujian fitur melihat laporan pemantauan progres mahasiswa	
Tujuan Pengujian	Menguji fitur melihat laporan pemantauan progres mahasiswa	
Skenario	Akun yang digunakan adalah dosen	
Kondisi Awal	Pengguna telah login dan berada di halaman beranda	
Langkah Pengujian	Respon Sistem yang Diharapkan	Berhasil
Pengguna memilih menu Progres Mahasiswa	Sistem menampilkan halaman Kelola Progres Mahasiswa	√
Pengguna menekan tombol Monitoring	Sistem menampilkan halaman laporan pemantauan progres mahasiswa	√
Hasil Akhir yang Diharapkan	Sistem menampilkan laporan pemantauan progres mahasiswa	
Hasil Akhir yang Didapat	Sistem dapat menampilkan laporan pemantauan progres mahasiswa	
Hasil Pengujian	Berhasil	

#	NRP	Nama	Bab 1	Bab 2	Bab 3	Total
1	5111100098	Ifan Iqbal	100	0	0	33.33%

Gambar 5.10 Hasil Pengujian Fitur Melihat Laporan Pemantauan Progres Mahasiswa

5.2.1.6 Pengujian Fitur Melakukan Verifikasi Progres

Pengujian ini adalah melakukan verifikasi progres melalui akun dosen. Rincian pengujian dapat dilihat pada Tabel 5.6, sedangkan hasil pengujian dapat dilihat pada Gambar 5.11.

Tabel 5.6 Pengujian Fitur Melakukan Verifikasi Progres

ID	FT-06	
Referensi Kasus Penggunaan	FR-06	
Nama	Pengujian fitur melakukan verifikasi progres	
Tujuan Pengujian	Menguji fitur melakukan verifikasi progres	
Skenario	Akun yang digunakan adalah dosen	
Kondisi Awal	Pengguna telah login dan berada di halaman beranda; telah ada data progres yang dimasukkan oleh mahasiswa bimbingan dosen	
Langkah Pengujian	Respon Sistem yang Diharapkan	Berhasil
Pengguna memilih menu Progres Mahasiswa	Sistem menampilkan halaman Verifikasi Progres Bimbingan	✓
Pengguna menekan tombol Verifikasi pada progres yang dinyatakan valid	Sistem mengubah status progres menjadi valid	✓
Pengguna menekan tombol Tidak Valid pada progres yang dinyatakan tidak valid	Sistem mengubah status progres menjadi tidak valid	✓
Hasil Akhir yang Diharapkan	Sistem menampilkan halaman verifikasi dan mampu menyimpan verifikasi yang dilakukan oleh pengguna	
Hasil Akhir yang Didapat	Sistem dapat menampilkan halaman verifikasi dan menyimpan verifikasi dengan mengubah status progres yang telah dilakukan verifikasi	
Hasil Pengujian	Berhasil	

#	NRP	Nama	Tanggal	Kegiatan Progres	Aksi
1	5111100098	Ifan Iqbal	11-01-2016 ✓	Progres buku	✓ ✗
2	5111100098	Ifan Iqbal	11-01-2016 ✗	Bab 1 (100%): Contoh bab 1	✓ ✗

Gambar 5.11 Hasil Pengujian Fitur Melakukan Verifikasi Progres

5.2.1.7 Pengujian Fitur Melihat Jadwal Sidang Pribadi (Dosen)

Pengujian ini adalah melihat jadwal sidang melalui akun dosen. Rincian pengujian dapat dilihat pada Tabel 5.7, sedangkan hasil pengujian dapat dilihat pada Gambar 5.12.

Tabel 5.7 Pengujian Fitur Melihat Jadwal Sidang Pribadi (Dosen)

ID	FT-07	
Referensi Kasus Penggunaan	FR-07	
Nama	Pengujian fitur melihat jadwal sidang pribadi (dosen)	
Tujuan Pengujian	Menguji fitur melihat jadwal sidang pribadi (dosen)	
Skenario	Akun yang digunakan adalah dosen	
Kondisi Awal	Pengguna telah login dan berada di halaman beranda	
Langkah Pengujian	Respon Sistem yang Diharapkan	Berhasil
Pengguna memilih menu Jadwal Sidang	Sistem menampilkan halaman Jadwal Sidang	√
Hasil Akhir yang Diharapkan	Sistem menampilkan jadwal sidang dosen	
Hasil Akhir yang Didapat	Sistem dapat menampilkan jadwal sidang dosen	
Hasil Pengujian	Berhasil	

#	NRP	Nama	Jenis Sidang	Tanggal	Mulai	Selesai	Ruangan
1	5111100098	Han Iqbal	Sidang Akhir	11-01-2018	08:00:00	10:00:00	R. Sidang 1

Gambar 5.12 Hasil Pengujian Fitur Melihat Jadwal Sidang Pribadi (Dosen)

5.2.1.8 Pengujian Fitur Mengelola Jadwal Sidang

Pengujian ini meliputi penambahan, pengubahan, dan penghapusan jadwal sidang. Rincian pengujian dapat dilihat pada Tabel 5.8, sedangkan hasil pengujian dapat dilihat pada Gambar 5.13 hingga Gambar 5.15.

Tabel 5.8 Pengujian Fitur Mengelola Jadwal Sidang

ID	FT-08	
Referensi Kasus Penggunaan	FR-08	
Nama	Pengujian fitur mengelola jadwal sidang	
Tujuan Pengujian	Menguji fitur mengelola jadwal sidang	
Skenario	Akun yang digunakan adalah koordinator	
Kondisi Awal	Pengguna telah login dan berada di halaman beranda	
Langkah Pengujian	Respon Sistem yang Diharapkan	Berhasil
Pengguna memilih menu Kelola Sidang	Sistem menampilkan halaman Kelola Sidang	√
Pengguna menekan tombol Tambah Sidang	Sistem menampilkan halaman penambahan jadwal sidang	√
Pengguna mengisi isian data sidang kemudian menekan tombol Simpan	Sistem menyimpan jadwal sidang dan menampilkannya dalam daftar jadwal sidang	√
Pengguna menekan tombol Edit pada jadwal sidang yang hendak diubah	Sistem menampilkan halaman pengubahan jadwal sidang	√
Pengguna mengubah isian data dan menekan tombol Simpan	Sistem meyimpan perubahan data	√
Pengguna menekan tombol hapus pada jadwal sidang yang hendak dihapus	Sistem menghapus sidang yang dipilih	√
Hasil Akhir yang Diharapkan	Sistem menampilkan halaman dan melakukan proses penambahan, pengubahan, dan penghapusan jadwal sidang	
Hasil Akhir yang Didapat	Sistem dapat menampilkan halaman dan melakukan proses penambahan, pengubahan, dan penghapusan jadwal sidang	
Hasil Pengujian	Berhasil	

Tambah Sidang Baru

NRP Mahasiswa	<input type="text"/>
Tanggal Sidang	19-01-2016
Sesi Sidang	Satu
Ruang Sidang	R. Sidang 1
Jenis Sidang	Sidang Akhir
Dosen Penguji 1	Umi Laili Yuhana
Dosen penguji 2	Umi Laili Yuhana
<input type="button" value="Simpan"/>	

Gambar 5.13 Hasil Pengujian Fitur Penambahan Jadwal Sidang

Edit Jadwal Sidang

NRP Mahasiswa	5111100098
Tanggal Sidang	11-01-2016
Sesi Sidang	Satu
Ruang Sidang	R. Sidang 1
Jenis Sidang	Sidang Akhir
Dosen Penguji 1	Umi Laili Yuhana
Dosen penguji 2	Diana Purwitasari
<input type="button" value="Simpan"/>	

Gambar 5.14 Hasil Pengujian Fitur Pengubahan Jadwal Sidang

Data sidang berhasil disimpan.

Kelola Sidang

[+ Tambah Sidang](#)

No.	Nama Mahasiswa	Jenis Sidang	Tanggal	Mulai	Selesai	Ruangan	Action
1	Irfan Iqbal	Sidang Akhir	11-01-2016	08:00:00	10:00:00	R. Sidang 1	G R

Showing 1 to 1 of 1 entries

Previous [1](#) Next

Gambar 5.15 Hasil Pengujian Fitur Mengelola Jadwal Sidang

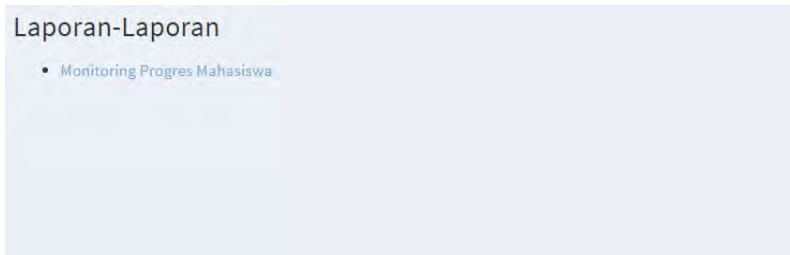
5.2.1.9 Pengujian Fitur Melihat Laporan-Laporan

Pengujian ini adalah melihat laporan-laporan yang tersedia untuk pengguna dengan peran sebagai koordinator. Rincian

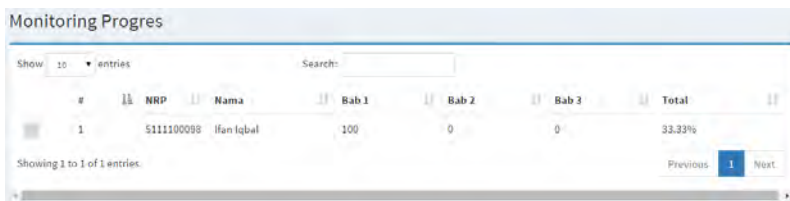
pengujian dapat dilihat pada Tabel 5.9, sedangkan hasil pengujian dapat dilihat pada Gambar 5.16 hingga Gambar 5.17.

Tabel 5.9 Pengujian Fitur Melihat Laporan-Laporan

ID	FT-09	
Referensi Kasus Penggunaan	FR-09	
Nama	Pengujian fitur melihat laporan-laporan	
Tujuan Pengujian	Menguji fitur melihat laporan-laporan	
Skenario	Akun yang digunakan adalah koordinator	
Kondisi Awal	Pengguna telah login dan berada di halaman beranda	
Langkah Pengujian	Respon Sistem yang Diharapkan	Berhasil
Pengguna memilih menu Laporan-Laporan	Sistem menampilkan halaman Laporan-Laporan	√
Pengguna memilih salah satu jenis laporan	Sistem menampilkan laporan yang dipilih	√
Hasil Akhir yang Diharapkan	Sistem menampilkan laporan yang dipilih pengguna	
Hasil Akhir yang Didapat	Sistem dapat menampilkan laporan yang dipilih pengguna	
Hasil Pengujian	Berhasil	



Gambar 5.16 Hasil Pengujian Fitur Melihat Laporan-Laporan



Gambar 5.17 Hasil Pengujian Fitur Melihat Laporan Monitoring Progres Seluruh Mahasiswa

5.2.1.10 Pengujian Fitur Mengelola Data Mahasiswa

Rincian pengujian dapat dilihat pada Tabel 5.10, sedangkan hasil pengujian dapat dilihat pada Gambar 5.18 dan Gambar 5.19.

Tabel 5.10 Pengujian Fitur Mengelola Data Mahasiswa

ID	FT-10	
Referensi Kasus Penggunaan	FR-10	
Nama	Pengujian fitur mengelola data mahasiswa	
Tujuan Pengujian	Menguji fitur mengelola data mahasiswa	
Skenario	Akun yang digunakan adalah administrator jurusan	
Kondisi Awal	Pengguna telah login dan berada di halaman beranda	
Langkah Pengujian	Respon Sistem yang Diharapkan	Berhasil
Pengguna memilih menu Kelola Mahasiswa	Sistem menampilkan halaman Kelola Mahasiswa	√
Pengguna menekan tombol Tambah Mahasiswa	Sistem menampilkan halaman penambahan mahasiswa	√
Pengguna mengisikan data mahasiswa, kemudian menekan tombol Simpan	Sistem menyimpan data mahasiswa dan menampilkan data yang baru tersimpan	√
Pengguna menekan tombol Edit pada data mahasiswa yang hendak diubah	Sistem menampilkan halaman perubahan data mahasiswa dan memuat data mahasiswa yang dipilih ke dalam isian perubahan data mahasiswa	√
Pengguna mengubah isian dan menekan tombol Simpan	Sistem menyimpan perubahan dan menampilkan data yang berhasil disimpan	√
Pengguna menekan tombol Hapus pada data yang hendak dihapus	Sistem menghapus data mahasiswa yang dipilih untuk dihapus	√
Hasil Akhir yang Diharapkan	Sistem menampilkan halaman dan melakukan proses penambahan, perubahan, dan penghapusan data mahasiswa	

Hasil Akhir yang Didapat	Sistem dapat menampilkan halaman dan melakukan proses penambahan, perubahan, dan penghapusan data mahasiswa
Hasil Pengujian	Berhasil

Tambah Mahasiswa

Nama

NRP

Jenjang Pendidikan ☐ D3 ☒ S1

Jenis Jenjang ☐ Reguler ☒ Lintas Jalur

Simpan

Gambar 5.18 Hasil Pengujian Fitur Penambahan Data Mahasiswa

Daftar Mahasiswa

[+ Tambah Mahasiswa](#)

No.	Nama	NRP	Jenjang	Jenis Jenjang	Action
1	Yudha	51111100090	S1	Reguler	 
2	Ifan Iqbal	51111100098	S1	Reguler	 

Gambar 5.19 Hasil Pengujian Fitur Melihat Halaman Kelola Mahasiswa

5.2.1.11 Pengujian Fitur Mengelola Nilai Tugas Akhir

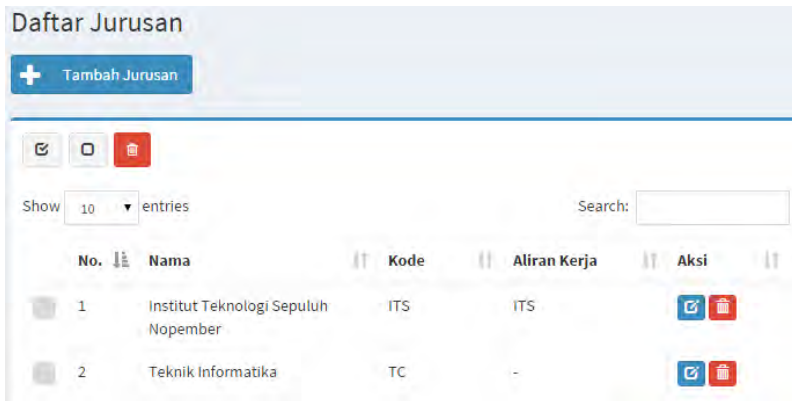
Pengujian ini meliputi fitur melihat daftar nilai dan memasukkan nilai yang dilakukan oleh administrator jurusan.

Rincian pengujian dapat dilihat pada Tabel 5.11, sedangkan hasil pengujian dapat dilihat pada Gambar 5.20 dan Gambar 5.21.

Tabel 5.11 Pengujian Fitur Mengelola Nilai Tugas Akhir

ID	FT-11	
Referensi Kasus Penggunaan	FR-11	
Nama	Pengujian fitur mengelola nilai tugas akhir	
Tujuan Pengujian	Menguji fitur mengelola nilai tugas akhir	
Skenario	Akun yang digunakan adalah administrator jurusan	
Kondisi Awal	Pengguna telah login dan berada di halaman beranda	
Langkah Pengujian	Respon Sistem yang Diharapkan	Berhasil
Pengguna memilih menu Kelola Nilai	Sistem menampilkan halaman Kelola Nilai	√
Pengguna menekan tombol Ubah Nilai pada baris mahasiswa tertentu	Sistem menampilkan halaman pengubahan nilai mahasiswa	√
Pengguna mengisikan nilai kemudian menekan tombol Simpan	Sistem menyimpan nilai tugas akhir mahasiswa	√
Hasil Akhir yang Diharapkan	Sistem menampilkan halaman daftar nilai dan mampu menangani proses pengisian nilai tugas akhir mahasiswa	
Hasil Akhir yang Didapat	Sistem menampilkan halaman daftar nilai dan mampu menangani proses pengisian nilai tugas akhir mahasiswa	
Hasil Pengujian	Berhasil	

Referensi Kasus Penggunaan	FR-12	
Nama	Pengujian fitur pengelolaan jurusan	
Tujuan Pengujian	Menguji fitur pengelolaan jurusan	
Skenario	Akun yang digunakan adalah administrator pusat.	
Kondisi Awal	Pengguna telah login dan berada di halaman beranda	
Langkah Pengujian	Respon Sistem yang Diharapkan	Berhasil
Pengguna memilih menu Kelola Jurusan	Sistem menampilkan halaman Kelola Jurusan	√
Pengguna memilih menu Tambah Jurusan	Sistem menampilkan halaman Tambah Jurusan	√
Pengguna memilih mengisi isian dan menekan tombol Simpan	Sistem menyimpan dan menampilkan pesan bahwa penyimpanan berhasil	√
Pengguna memilih menu Sunting Jurusan	Sistem menampilkan halaman Ubah Jurusan	√
Pengguna mengubah isian dan menekan tombol Simpan	Sistem menyimpan perubahan dan menampilkan pesan bahwa perubahan telah disimpan	√
Pengguna menekan tombol Hapus untuk menghapus jurusan yang dipilih	Sistem menghapus jurusan dan menampilkan pesan bahwa penghapusan berhasil	√
Hasil Akhir yang Diharapkan	Sistem menampilkan daftar jurusan dan mampu menangani pengelolaan data jurusan seperti penambahan, pembaruan, dan penghapusan data.	
Hasil Akhir yang Didapat	Sistem dapat menampilkan daftar data jurusan hasil pengelolaan data.	
Hasil Pengujian	Berhasil	



Gambar 5.22 Hasil Pengujian Fitur Mengelola Jurusan

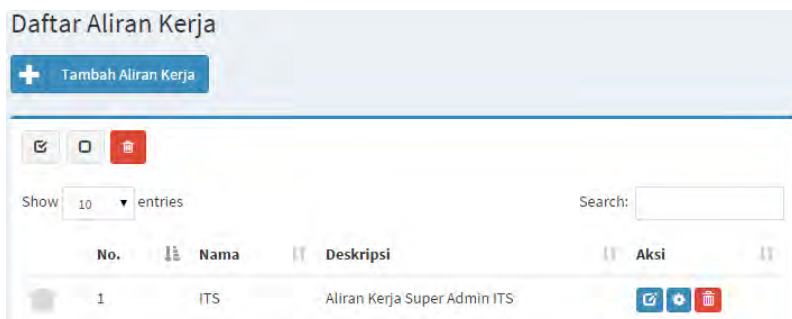
5.2.1.13 Pengujian Fitur Mengelola Aliran Kerja

Pengujian fitur ini adalah menambah data, memperbarui, dan menghapus data aliran kerja. Selain itu, terdapat pengujian terhadap subfitur pemetaan proses pada aliran kerja. Rincian pengujian dapat dilihat pada Tabel 5.13, sedangkan hasil pengujian dapat dilihat pada Gambar 5.23.

Tabel 5.13 Pengujian Fitur Mengelola Aliran Kerja

ID	FT-13	
Referensi Kasus Penggunaan	FR-13	
Nama	Pengujian fitur pengelolaan aliran kerja	
Tujuan Pengujian	Menguji fitur pengelolaan aliran kerja	
Skenario	Akun yang digunakan adalah administrator pusat.	
Kondisi Awal	Pengguna telah login dan berada di halaman beranda	
Langkah Pengujian	Respon Sistem yang Diharapkan	Berhasil
Pengguna memilih menu Kelola aliran kerja	Sistem menampilkan halaman Kelola aliran kerja	√
Pengguna memilih menu Tambah aliran kerja	Sistem menampilkan halaman Tambah aliran kerja	√

Pengguna memilih mengisi isian dan menekan tombol Simpan	Sistem menyimpan dan menampilkan pesan bahwa penyimpanan berhasil	✓
Pengguna memilih menu Sunting aliran kerja	Sistem menampilkan halaman Ubah aliran kerja	✓
Pengguna mengubah isian dan menekan tombol Simpan	Sistem menyimpan perubahan dan menampilkan pesan bahwa perubahan telah disimpan	✓
Pengguna menekan tombol Hapus untuk menghapus aliran kerja yang dipilih	Sistem menghapus aliran kerja dan menampilkan pesan bahwa penghapusan berhasil	✓
Pengguna menekan tombol Pemetaan Proses untuk memetakan proses	Sistem menampilkan halaman pemetaan proses	✓
Pengguna mengisi isian pemetaan proses kemudian menekan tombol Simpan	Sistem menyimpan pemetaan proses dan menampilkan pesan bahwa pemetaan berhasil disimpan	✓
Hasil Akhir yang Diharapkan	Sistem menampilkan daftar aliran kerja dan mampu menangani pengelolaan data aliran kerja seperti penambahan, pembaruan, dan penghapusan data.	
Hasil Akhir yang Didapat	Sistem dapat menampilkan daftar data aliran kerja hasil pengelolaan data.	
Hasil Pengujian	Berhasil	



Gambar 5.23 Hasil Pengujian Fitur Mengelola Aliran Kerja

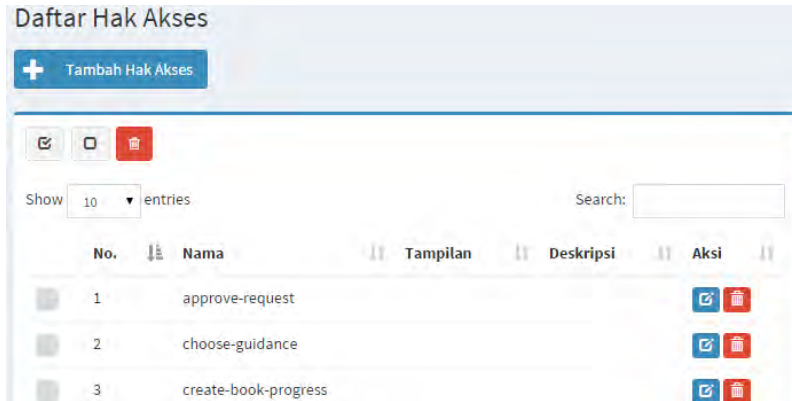
5.2.1.14 Pengujian Fitur Mengelola Hak Akses

Pengujian fitur ini adalah menambah data, memperbarui, dan menghapus data hak akses. Rincian pengujian dapat dilihat pada Tabel 5.14, sedangkan hasil pengujian dapat dilihat pada Gambar 5.24.

Tabel 5.14 Pengujian Fitur Mengelola Hak Akses

ID	FT-14	
Referensi Kasus Penggunaan	FR-14	
Nama	Pengujian fitur pengelolaan hak akses	
Tujuan Pengujian	Menguji fitur pengelolaan hak akses	
Skenario	Akun yang digunakan adalah administrator pusat.	
Kondisi Awal	Pengguna telah login dan berada di halaman beranda	
Langkah Pengujian	Respon Sistem yang Diharapkan	Berhasil
Pengguna memilih menu Kelola hak akses	Sistem menampilkan halaman Kelola hak akses	√
Pengguna memilih menu Tambah hak akses	Sistem menampilkan halaman Tambah hak akses	√
Pengguna memilih mengisi isian dan menekan tombol Simpan	Sistem menyimpan dan menampilkan pesan bahwa penyimpanan berhasil	√
Pengguna memilih menu Sunting hak akses	Sistem menampilkan halaman Ubah hak akses	√
Pengguna mengubah isian dan menekan tombol Simpan	Sistem menyimpan perubahan dan menampilkan pesan bahwa perubahan telah disimpan	√
Pengguna menekan tombol Hapus untuk menghapus hak akses yang dipilih	Sistem menghapus hak akses dan menampilkan pesan bahwa penghapusan berhasil	√
Hasil Akhir yang Diharapkan	Sistem menampilkan daftar hak akses dan mampu menangani pengelolaan data hak akses seperti penambahan, pembaruan, dan penghapusan data.	

Hasil Akhir yang Didapat	Sistem dapat menampilkan daftar data hak akses hasil pengelolaan data.
Hasil Pengujian	Berhasil



Gambar 5.24 Hasil Pengujian Fitur Mengelola Hak Akses

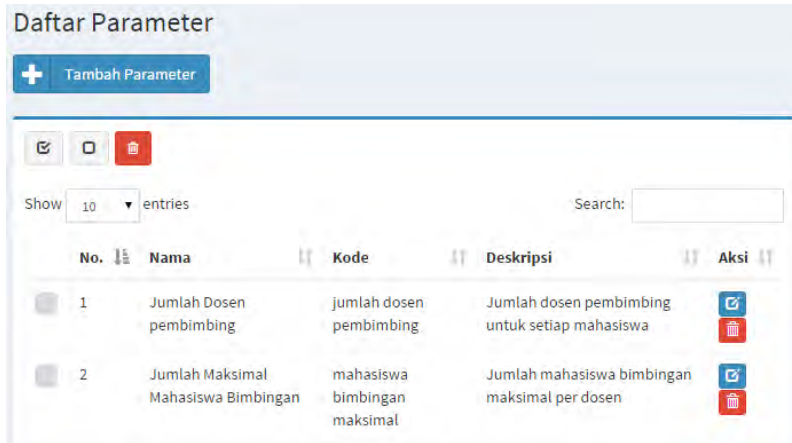
5.2.1.15 Pengujian Fitur Mengelola Parameter

Pengujian fitur ini adalah menambah data, memperbarui, dan menghapus data parameter. Rincian pengujian dapat dilihat pada Tabel 5.15, sedangkan hasil pengujian dapat dilihat pada Gambar 5.25.

Tabel 5.15 Pengujian Fitur Mengelola Parameter

ID	FT-15	
Referensi Kasus Penggunaan	FR-15	
Nama	Pengujian fitur pengelolaan parameter	
Tujuan Pengujian	Menguji fitur pengelolaan parameter	
Skenario	Akun yang digunakan adalah administrator pusat.	
Kondisi Awal	Pengguna telah login dan berada di halaman beranda	
Langkah Pengujian	Respon Sistem yang Diharapkan	Berhasil
Pengguna memilih menu Kelola parameter	Sistem menampilkan halaman Kelola parameter	√
Pengguna memilih menu Tambah parameter	Sistem menampilkan halaman Tambah parameter	√
Pengguna memilih mengisi isian dan menekan tombol Simpan	Sistem menyimpan dan menampilkan pesan bahwa penyimpanan berhasil	√
Pengguna memilih menu Sunting parameter	Sistem menampilkan halaman Ubah parameter	√
Pengguna mengubah isian dan menekan tombol Simpan	Sistem menyimpan perubahan dan menampilkan pesan bahwa perubahan telah disimpan	√
Pengguna menekan tombol Hapus untuk menghapus parameter yang dipilih	Sistem menghapus parameter dan menampilkan pesan bahwa penghapusan berhasil	√
Hasil Akhir yang Diharapkan	Sistem menampilkan daftar parameter dan mampu menangani pengelolaan data parameter seperti penambahan, pembaruan, dan penghapusan data.	

Hasil Akhir yang Didapat	Sistem dapat menampilkan daftar data parameter hasil pengelolaan data.
Hasil Pengujian	Berhasil



Gambar 5.25 Hasil Pengujian Fitur Mengelola Parameter

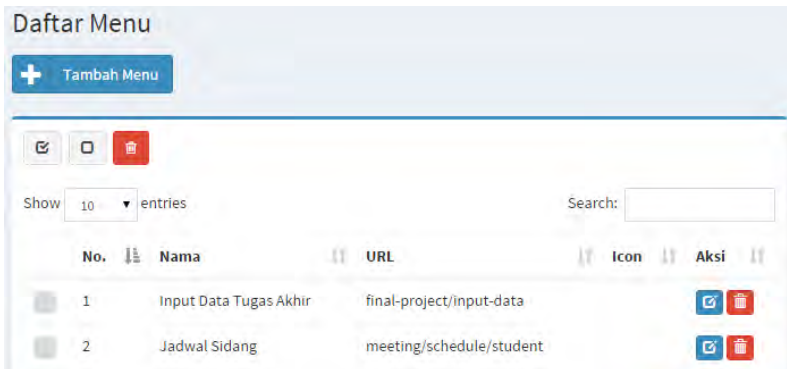
5.2.1.16 Pengujian Fitur Mengelola Menu

Pengujian fitur ini adalah menambah data, memperbarui, dan menghapus data menu. Rincian pengujian dapat dilihat pada Tabel 5.16, sedangkan hasil pengujian dapat dilihat pada Gambar 5.26.

Tabel 5.16 Pengujian Fitur Mengelola Menu

ID	FT-16	
Referensi Kasus Penggunaan	FR-16	
Nama	Pengujian fitur pengelolaan menu	
Tujuan Pengujian	Menguji fitur pengelolaan menu	
Skenario	Akun yang digunakan adalah administrator pusat.	
Kondisi Awal	Pengguna telah login dan berada di halaman beranda	
Langkah Pengujian	Respon Sistem yang Diharapkan	Berhasil
Pengguna memilih menu Kelola menu	Sistem menampilkan halaman Kelola menu	√
Pengguna memilih menu Tambah menu	Sistem menampilkan halaman Tambah menu	√

Pengguna memilih mengisi isian dan menekan tombol Simpan	Sistem menyimpan dan menampilkan pesan bahwa penyimpanan berhasil	√
Pengguna memilih menu Sunting menu	Sistem menampilkan halaman Ubah menu	√
Pengguna mengubah isian dan menekan tombol Simpan	Sistem menyimpan perubahan dan menampilkan pesan bahwa perubahan telah disimpan	√
Pengguna menekan tombol Hapus untuk menghapus menu yang dipilih	Sistem menghapus menu dan menampilkan pesan bahwa penghapusan berhasil	√
Hasil Akhir yang Diharapkan	Sistem menampilkan daftar menu dan mampu menangani pengelolaan data menu seperti penambahan, pembaruan, dan penghapusan data.	
Hasil Akhir yang Didapat	Sistem dapat menampilkan daftar data menu hasil pengelolaan data.	
Hasil Pengujian	Berhasil	



Gambar 5.26 Hasil Pengujian Fitur Mengelola Menu

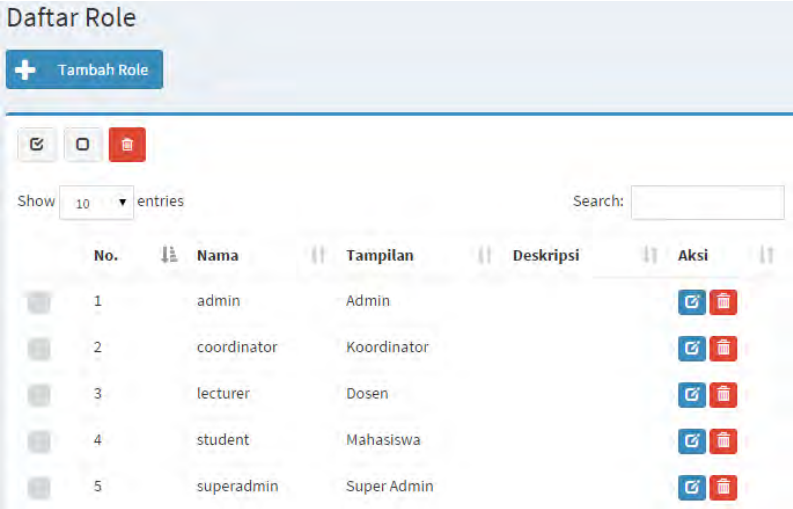
5.2.1.17 Pengujian Fitur Mengelola Peran

Pengujian fitur ini adalah menambah data, memperbarui, dan menghapus data peran. Rincian pengujian dapat dilihat pada

Tabel 5.17, sedangkan hasil pengujian dapat dilihat pada Gambar 5.27.

Tabel 5.17 Pengujian Fitur Mengelola Peran

ID	FT-17	
Referensi Kasus Penggunaan	FR-17	
Nama	Pengujian fitur pengelolaan peran	
Tujuan Pengujian	Menguji fitur pengelolaan peran	
Skenario	Akun yang digunakan adalah administrator pusat.	
Kondisi Awal	Pengguna telah login dan berada di halaman beranda	
Langkah Pengujian	Respon Sistem yang Diharapkan	Berhasil
Pengguna memilih menu Kelola peran	Sistem menampilkan halaman Kelola peran	√
Pengguna memilih menu Tambah menu	Sistem menampilkan halaman Tambah menu	√
Pengguna memilih mengisi isian dan menekan tombol Simpan	Sistem menyimpan dan menampilkan pesan bahwa penyimpanan berhasil	√
Pengguna memilih menu Sunting peran	Sistem menampilkan halaman Ubah peran	√
Pengguna mengubah isian dan menekan tombol Simpan	Sistem menyimpan perubahan dan menampilkan pesan bahwa perubahan telah disimpan	√
Pengguna menekan tombol Hapus untuk menghapus peran yang dipilih	Sistem menghapus peran dan menampilkan pesan bahwa penghapusan berhasil	√
Hasil Akhir yang Diharapkan	Sistem menampilkan daftar peran dan mampu menangani pengelolaan data peran seperti penambahan, pembaruan, dan penghapusan data.	
Hasil Akhir yang Didapat	Sistem dapat menampilkan daftar data peran hasil pengelolaan data.	
Hasil Pengujian	Berhasil	



Gambar 5.27 Hasil Pengujian Fitur Mengelola Peran

5.2.1.18 Pengujian Aliran Kerja Jurusan

Pengujian ini dilakukan untuk menguji fitur pengelolaan aliran kerja terhadap prosedur tugas akhir dari jurusan. Rincian pengujian dapat dilihat pada Tabel 5.18, sedangkan hasil pengujian dapat dilihat pada Gambar 5.28 dan Gambar 5.29.

Tabel 5.18 Pengujian Aliran Kerja Jurusan

ID	FT-18	
Referensi Kasus Penggunaan	FR-13, FR-14, FR-15, FR-16, dan FR-17	
Nama	Pengujian aliran kerja jurusan	
Tujuan Pengujian	Menguji aliran kerja jurusan	
Skenario	Akun yang digunakan adalah administrator pusat.	
Kondisi Awal	Pengguna telah login dan berada di halaman beranda	
Langkah Pengujian	Respon Sistem yang Diharapkan	Berhasil
Pengguna membuka halaman Kelola Jurusan kemudian	Sistem menampilkan halaman kelola jurusan dan menyimpan jurusan yang ditambahkan	√

menambahkan Jurusan Teknik Informatika		
Pengguna membuka halaman Kelola Aliran Kerja kemudian menambahkan aliran kerja Teknik Informatika	Sistem menampilkan halaman kelola aliran kerja dan menyimpan aliran kerja yang ditambahkan	√
Pengguna membuka halaman Kelola Aliran Kerja kemudian menekan tombol Mapping Process	Sistem menampilkan halaman kelola aliran kerja dan menampilkan halaman <i>mapping process</i>	√
Pengguna menekan tombol Tambah Proses kemudian memilih peran dan menu untuk setiap proses yang ditambahkan	Sistem menampilkan jendela pemetaan peran terhadap menu	√
Pengguna menekan tombol Simpan	Sistem menyimpan proses-proses yang sudah dipetakan	√
Hasil Akhir yang Diharapkan	Sistem menampilkan halaman dan menangani proses pengelolaan aliran kerja jurusan	
Hasil Akhir yang Didapat	Sistem dapat menampilkan halaman dan menangani proses pengelolaan aliran kerja jurusan	
Hasil Pengujian	Berhasil	

Tambah Jurusan

Nama

Teknik Informatika

Kode

TC

Aliran Kerja

Teknik Informatika

Simpan

Gambar 5.28 Hasil Pengujian Penambahan Jurusan

Pemetaan Proses Aliran Kerja

Aliran Kerja Teknik Informatika

Tambah Proses

Peran	Menu	Aksi
Admin	Kelola Mahasiswa	
Admin	Kelola Nilai	
Mahasiswa	Pilih Dosen Pembimbing	
Mahasiswa	Unggah Proposal	
Mahasiswa	Input Data Tugas Akhir	
Mahasiswa	Progres Bimbingan	
Mahasiswa	Unggah Skripsi	

Gambar 5.29 Hasil Pengujian Pemetaan Proses Aliran Kerja

5.3 Evaluasi Pengujian Fungsionalitas

Pada subbab ini akan diberikan hasil evaluasi dari pengujian-pengujian yang telah dilakukan. Evaluasi yang diberikan meliputi evaluasi pengujian kebutuhan fungsional.

5.3.1 Evaluasi Pengujian Fungsionalitas

Rangkuman mengenai hasil pengujian fungsionalitas dapat dilihat pada Tabel 5.19. Berdasarkan data pada tabel tersebut, semua skenario pengujian berhasil dan program berjalan dengan baik. Sehingga bisa ditarik disimpulkan bahwa fungsionalitas dari program telah bisa bekerja sesuai dengan yang diharapkan.

Tabel 5.19 Rangkuman Hasil Pengujian Fungsionalitas

ID	Nama	Hasil
FT-01	Pengujian fitur memilih dosen pembimbing	Berhasil
FT-02	Pengujian fitur mengunggah berkas proposal	Berhasil
FT-03	Pengujian fitur mengelola progres bimbingan dan progres buku	Berhasil
FT-04	Pengujian fitur melihat jadwal sidang pribadi (mahasiswa)	Berhasil
FT-05	Pengujian fitur melihat laporan pemantauan progres mahasiswa	Berhasil
FT-06	Pengujian fitur melakukan verifikasi progres	Berhasil
FT-07	Pengujian fitur melihat jadwal sidang pribadi (dosen)	Berhasil
FT-08	Pengujian fitur mengelola jadwal sidang	Berhasil
FT-09	Pengujian fitur melihat laporan-laporan	Berhasil
FT-10	Pengujian fitur mengelola data mahasiswa	Berhasil
FT-11	Pengujian fitur mengelola nilai tugas akhir	Berhasil
FT-12	Pengujian fitur mengelola nilai jurusan	Berhasil
FT-13	Pengujian fitur mengelola nilai aliran kerja	Berhasil
FT-14	Pengujian fitur mengelola nilai hak akses	Berhasil
FT-15	Pengujian fitur mengelola nilai parameter	Berhasil
FT-16	Pengujian fitur mengelola nilai menu	Berhasil
FT-17	Pengujian fitur mengelola nilai peran	Berhasil
FT-18	Pengujian aliran kerja jurusan	Berhasil

BAB 6

KESIMPULAN DAN SARAN

Pada bab ini akan diberikan kesimpulan yang diambil selama pengerjaan tugas akhir serta saran-saran tentang pengembangan yang dapat dilakukan terhadap tugas akhir ini di masa yang akan datang.

6.1 Kesimpulan

Dari hasil selama proses perancangan, implementasi, serta pengujian dapat diambil kesimpulan sebagai berikut.

1. Aplikasi Sistem Informasi Administrasi Tugas Akhir ITS mampu mengelola dan menjalankan prosedur tugas akhir dari beberapa jurusan melalui fitur pengaturan aliran kerja yang telah diimplementasikan.
2. Aplikasi Sistem Informasi Administrasi Tugas Akhir ITS dapat menerima perbedaan-perbedaan parameter antar beberapa jurusan dan menyimpannya dalam basis data melalui fitur pengaturan parameter jurusan.
3. Aplikasi Sistem Informasi Administrasi Tugas Akhir ITS mampu menyediakan kebutuhan administrasi tugas akhir yang generik untuk jurusan-jurusan di ITS.

6.2 Saran

Berikut saran-saran untuk pengembangan dan perbaikan sistem di masa yang akan datang, di antaranya adalah sebagai berikut.

1. Pemetaan proses pada pengaturan aliran kerja dapat lebih dimaksimalkan dengan merancang antarmuka yang memiliki interaktivitas pengguna yang lebih baik.
2. Beberapa logika program yang terdapat di beberapa pengendali (*controller*) dapat diekstrak ke dalam sebuah layanan (*service*) tersendiri agar kode program memiliki atribut *reusability* yang lebih baik.
3. Terdapat modul pengumuman atau berita yang dapat dilihat oleh pengguna umum sebagai media komunikasi antara pihak-

pihak yang berkepentingan dalam pelaksanaan prosedur tugas akhir di jurusan.

LAMPIRAN A. IMPLEMENTASI APLIKASI

1. Kode Program Implementasi Migrasi Basis Data

```
class InitialSchema extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('users', function (Blueprint $table) {
            $table->increments('id');
            $table->string('name');
            $table->string('username')->unique();
            $table->string('password', 60);
            $table->unsignedInteger('role_id');
            $table->rememberToken();
            $table->timestamps();

            $table->unsignedInteger('person_id')->index();
            $table->string('person_type')->index();
            $table->unsignedInteger('major_id')->index();
        });

        Schema::create('students', function (Blueprint $table) {
            $table->increments('id');
            $table->unsignedInteger('major_id')->index();
            $table->unsignedInteger('education_degree_id')->index();
            $table->unsignedInteger('degree_type_id')->index();
            $table->string('nrp')->unique();
            $table->string('name');
            $table->timestamps();
        });
    }
}
```

```

        $table->unsignedInteger('project_id')->nullable();
    });

    Schema::create('lecturers', function (Blueprint $table) {
        $table->increments('id');
        $table->unsignedInteger('major_id')->index();
        $table->string('nip')->unique();
        $table->string('name');
        $table->timestamps();
    });

    Schema::create('external_lecturers', function (Blueprint
$table) {
        $table->increments('id');
        $table->string('code')->index();
        $table->string('name');
        $table->timestamps();
    });

    Schema::create('proposals', function (Blueprint $table) {
        $table->increments('id');
        $table->text('title');
        $table->string('file');
        $table->text('abstract');
        $table->unsignedInteger('student_id');
        $table->unsignedInteger('project_id');
        $table->unsignedInteger('major_id');
        $table->timestamps();
    });

    Schema::create('progresses', function (Blueprint $table) {
        $table->increments('id');
        $table->text('activity');
        $table->date('event');
        $table->unsignedInteger('status_id')->index();
        $table->unsignedInteger('project_id')->index();
        $table->unsignedInteger('student_id')->index();
        $table->unsignedInteger('major_id')->index();
        $table->timestamps();
    });

    Schema::create('book_progresses', function (Blueprint
$table) {
        $table->increments('id');
        $table->unsignedInteger('chapter_id')->index();
        $table->float('performance');
        $table->string('file');
        $table->text('explanation');
        $table->date('event');
    });

```

```

        $table->unsignedInteger('status_id')->index();
        $table->unsignedInteger('project_id')->index();
        $table->unsignedInteger('student_id')->index();
        $table->unsignedInteger('major_id')->index();
        $table->timestamps();
    });

    Schema::create('progress_statuses', function (Blueprint
$table) {
        $table->increments('id');
        $table->string('code')->nullable();
        $table->string('name');
    });

    Schema::create('meetings', function (Blueprint $table) {
        $table->increments('id');
        $table->unsignedInteger('session_id')->index();
        $table->unsignedInteger('room_id')->index();
        $table->unsignedInteger('project_id')->index();
        $table->unsignedInteger('student_id')->index();
        $table->unsignedInteger('type_id')->index();
        $table->unsignedInteger('major_id')->index();
        $table->date('held');
        $table->timestamps();
    });

    Schema::create('rooms', function (Blueprint $table) {
        $table->increments('id');
        $table->string('code');
        $table->string('name');
        $table->unsignedInteger('major_id')->index();
        $table->timestamps();
    });

    Schema::create('education_degrees', function (Blueprint
$table) {
        $table->increments('id');
        $table->string('code');
        $table->string('name');
        $table->unsignedInteger('major_id')->index();
        $table->timestamps();
    });

    Schema::create('degree_types', function (Blueprint $table)
{
        $table->increments('id');
        $table->string('code');
        $table->string('name');
        $table->unsignedInteger('major_id')->index();

```

```

        $table->timestamps();
    });

    Schema::create('majors', function (Blueprint $table) {
        $table->increments('id');
        $table->string('code');
        $table->string('name');
        $table->unsignedInteger('workflow_id')->index();
        $table->timestamps();
    });

    Schema::create('interest_areas', function (Blueprint
$table) {
        $table->increments('id');
        $table->string('code');
        $table->string('name');
        $table->unsignedInteger('major_id')->index();
        $table->timestamps();
    });

    Schema::create('projects', function (Blueprint $table) {
        $table->increments('id');
        $table->text('title');
        $table->date('start')->nullable();
        $table->string('file');
        $table->text('abstract');
        $table->float('proposal_grade');
        $table->float('final_grade');
        $table->float('average_grade');
        $table->timestamps();
        $table->unsignedInteger('student_id')->index();
        $table->unsignedInteger('interest_area_id')->index();
        $table->unsignedInteger('major_id')->index();
    });

    Schema::create('meeting_types', function (Blueprint
$table) {
        $table->increments('id');
        $table->string('code');
        $table->string('name');
        $table->timestamps();
    });

    Schema::create('meeting_sessions', function (Blueprint
$table) {
        $table->increments('id');
        $table->string('code');
        $table->string('name');
        $table->time('start');
    });

```

```

        $table->time('finish');
        $table->unsignedInteger('major_id')->index();
        $table->timestamps();
    });

    Schema::create('chapters', function (Blueprint $table) {
        $table->increments('id');
        $table->string('code')->nullable();
        $table->string('name');
        $table->unsignedInteger('major_id')->index();
        $table->timestamps();
    });

    Schema::create('lecturer_project', function (Blueprint
$table) {
        $table->unsignedInteger('lecturer_id');
        $table->unsignedInteger('project_id');

        $table->primary(['lecturer_id', 'project_id']);
    });

    Schema::create('parameters', function (Blueprint $table) {
        $table->increments('id');
        $table->string('code');
        $table->string('name');
        $table->string('description');
        $table->timestamps();
    });

    Schema::create('major_parameter', function (Blueprint
$table) {
        $table->unsignedInteger('major_id');
        $table->unsignedInteger('parameter_id');
        $table->string('value');

        $table->primary(['major_id', 'parameter_id']);
    });

    Schema::create('workflows', function (Blueprint $table) {
        $table->increments('id');
        $table->string('name');
        $table->string('description');
        $table->timestamps();
    });

    Schema::create('lecturer_meeting', function (Blueprint
$table) {
        $table->unsignedInteger('lecturer_id');
        $table->unsignedInteger('meeting_id');

```

```

        $table->foreign('lecturer_id')->references('id')->
>on('lecturers')
        ->onUpdate('cascade')->onDelete('cascade');
        $table->foreign('meeting_id')->references('id')->
>on('meetings')
        ->onUpdate('cascade')->onDelete('cascade');
    });

    // foreign key dikoding di sini
    Schema::table('proposals', function (Blueprint $table) {
        $table->foreign('student_id')->references('id')->
>on('students')
        ->onUpdate('cascade')->onDelete('cascade');
        $table->foreign('project_id')->references('id')->
>on('projects')
        ->onUpdate('cascade')->onDelete('cascade');
        $table->foreign('major_id')->references('id')->
>on('majors')
        ->onUpdate('cascade')->onDelete('cascade');
    });

    Schema::table('projects', function (Blueprint $table) {
        $table->foreign('student_id')->references('id')->
>on('students')
        ->onUpdate('cascade')->onDelete('cascade');
        $table->foreign('interest_area_id')->references('id')->
>on('interest_areas')
        ->onUpdate('cascade')->onDelete('cascade');
        $table->foreign('major_id')->references('id')->
>on('majors')
        ->onUpdate('cascade')->onDelete('cascade');
    });

    Schema::table('lecturer_project', function (Blueprint
$table) {
        $table->foreign('lecturer_id')->references('id')->
>on('lecturers')
        ->onUpdate('cascade')->onDelete('cascade');
        $table->foreign('project_id')->references('id')->
>on('projects')
        ->onUpdate('cascade')->onDelete('cascade');
    });

    Schema::table('progresses', function (Blueprint $table) {
        $table->foreign('status_id')->references('id')->
>on('progress_statuses')
        ->onUpdate('cascade')->onDelete('cascade');
    });

```

```

        $table->foreign('student_id')->references('id')->
>on('students')
        ->onUpdate('cascade')->onDelete('cascade');
        $table->foreign('project_id')->references('id')->
>on('projects')
        ->onUpdate('cascade')->onDelete('cascade');
        $table->foreign('major_id')->references('id')->
>on('majors')
        ->onUpdate('cascade')->onDelete('cascade');
    });

    Schema::table('book_progresses', function (Blueprint
$table) {
        $table->foreign('status_id')->references('id')->
>on('progress_statuses')
        ->onUpdate('cascade')->onDelete('cascade');
        $table->foreign('student_id')->references('id')->
>on('students')
        ->onUpdate('cascade')->onDelete('cascade');
        $table->foreign('project_id')->references('id')->
>on('projects')
        ->onUpdate('cascade')->onDelete('cascade');
        $table->foreign('major_id')->references('id')->
>on('majors')
        ->onUpdate('cascade')->onDelete('cascade');
    });

    Schema::table('meetings', function (Blueprint $table) {
        $table->foreign('room_id')->references('id')->
>on('rooms')
        ->onUpdate('cascade')->onDelete('cascade');
        $table->foreign('student_id')->references('id')->
>on('students')
        ->onUpdate('cascade')->onDelete('cascade');
        $table->foreign('project_id')->references('id')->
>on('projects')
        ->onUpdate('cascade')->onDelete('cascade');
        $table->foreign('major_id')->references('id')->
>on('majors')
        ->onUpdate('cascade')->onDelete('cascade');
        $table->foreign('type_id')->references('id')->
>on('meeting_types')
        ->onUpdate('cascade')->onDelete('cascade');
    });

    Schema::table('rooms', function (Blueprint $table) {
        $table->foreign('major_id')->references('id')->
>on('majors')
        ->onUpdate('cascade')->onDelete('cascade');
    });

```

```

        Schema::table('education_degrees', function (Blueprint
$table) {
            $table->foreign('major_id')->references('id')-
>on('majors')
                ->onUpdate('cascade')->onDelete('cascade');
        });
        Schema::table('degree_types', function (Blueprint $table)
{
            $table->foreign('major_id')->references('id')-
>on('majors')
                ->onUpdate('cascade')->onDelete('cascade');
        });
        Schema::table('interest_areas', function (Blueprint
$table) {
            $table->foreign('major_id')->references('id')-
>on('majors')
                ->onUpdate('cascade')->onDelete('cascade');
        });
        Schema::table('chapters', function (Blueprint $table) {
            $table->foreign('major_id')->references('id')-
>on('majors')
                ->onUpdate('cascade')->onDelete('cascade');
        });
        Schema::table('users', function (Blueprint $table) {
            $table->foreign('major_id')->references('id')-
>on('majors')
                ->onUpdate('cascade')->onDelete('cascade');
        });
    }

    /**
     * Reverse the migrations.
     *
     *
     */
    public function down()
    {
        DB::statement('SET FOREIGN_KEY_CHECKS = 0');
        Schema::drop('users');
        Schema::drop('students');
        Schema::drop('lecturers');
        Schema::drop('external_lecturers');
        Schema::drop('proposals');
        Schema::drop('progresses');
        Schema::drop('book_progresses');
        Schema::drop('progress_statuses');
        Schema::drop('meetings');
        Schema::drop('rooms');
    }

```



```

        Schema::drop('education_degrees');
        Schema::drop('degree_types');
        Schema::drop('majors');
        Schema::drop('interest_areas');
        Schema::drop('projects');
        Schema::drop('meeting_types');
        Schema::drop('meeting_sessions');
        Schema::drop('chapters');
        Schema::drop('lecturer_project');
        Schema::drop('parameters');
        Schema::drop('major_parameter');
        Schema::drop('workflows');
        Schema::drop('lecturer_meeting');
        DB::statement('SET FOREIGN_KEY_CHECKS = 1');
    }
}

```

Kode Sumber A.1 Migrasi Basis Data

2. Kode Program Implementasi Model BookProgress

```

<?php
namespace Simta\Models;

use Illuminate\Database\Eloquent\Model;
use Simta\Models\Traits\ProgresssetStatus;

class BookProgress extends Model
{
    use ProgresssetStatus;

    protected $table = 'book_progresses';

    protected $primaryKey = 'id';

    public $incrementing = true;

    public $timestamps = true;

    protected $dates = ['event'];

    protected $fillable = ['performance', 'file',
        'explanation'];

    const TYPE = 'book-progress';

    public function toString()
    {

```

```

        return '<a href="' . $this->file . '"><span class="fa
fa-file-o"></span> ' . $this->chapter->name . '</a>' . ' (' .
$this->performance . '%): ' . $this->explanation;
    }

    public function chapter()
    {
        return $this->belongsTo('Simta\Models\Chapter');
    }

    public function student()
    {
        return $this->belongsTo('Simta\Models\Student');
    }

    public function project()
    {
        return $this->belongsTo('Simta\Models\Project');
    }

    public function status()
    {
        return $this->belongsTo('Simta\Models\ProgressStatus',
'status_id');
    }

    public function major()
    {
        return $this->belongsTo('Simta\Models\Major');
    }
}

```

Kode Sumber A.2 Model BookProgress

3. Kode Program Implementasi Model Chapter

```

<?php

namespace Simta\Models;

use Illuminate\Database\Eloquent\Model;

class Chapter extends Model
{
    protected $table = 'chapters';

    protected $primaryKey = 'id';

    public $incrementing = true;
}

```

```

    public $timestamps = true;

    protected $fillable = ['code', 'name'];

    public function major()
    {
        return $this->belongsTo('Simta\Models\Major');
    }
}

```

Kode Sumber A.3 Model Chapter

4. Kode Program Implementasi Model DegreeType

```

<?php

namespace Simta\Models;

use Illuminate\Database\Eloquent\Model;

class DegreeType extends Model
{
    protected $table = 'degree_types';

    protected $primaryKey = 'id';

    public $incrementing = true;

    public $timestamps = true;

    protected $fillable = ['code', 'name'];

    public function major()
    {
        return $this->belongsTo('Simta\Models\Major');
    }
}

```

Kode Sumber A.4 Model DegreeType

5. Kode Program Implementasi Model EducationDegree

```

<?php

namespace Simta\Models;

use Illuminate\Database\Eloquent\Model;

class EducationDegree extends Model
{
    protected $table = 'education_degrees';
}

```

```

protected $primaryKey = 'id';

public $incrementing = true;

public $timestamps = true;

protected $fillable = ['code', 'name'];

public function major()
{
    return $this->belongsTo('Simta\Models\Major');
}

```

Kode Sumber A.5 Model EducationDegree

6. Kode Program Implementasi Model ExternalLecturer

```

<?php

namespace Simta\Models;

use Illuminate\Database\Eloquent\Model;

class ExternalLecturer extends Model
{
    protected $table = 'external_lecturers';

    protected $primaryKey = 'id';

    public $incrementing = true;

    public $timestamps = true;

    protected $fillable = ['code', 'name'];
}

```

Kode Sumber A.6 Model ExternalLecturer

7. Kode Program Implementasi Model InterestArea

```

<?php

namespace Simta\Models;

use Illuminate\Database\Eloquent\Model;

class InterestArea extends Model
{

```

```

        protected $table = 'interest_areas';

        protected $primaryKey = 'id';

        public $incrementing = true;

        public $timestamps = true;

        protected $fillable = ['code', 'name'];

        public function major()
        {
            return $this->belongsTo('Simta\Models\Major');
        }
    }

```

Kode Sumber A.7 Model InterestArea

8. Kode Program Implementasi Model Lecturer

```

<?php

namespace Simta\Models;

use Illuminate\Database\Eloquent\Model;

class Lecturer extends Model
{
    protected $table = 'lecturers';

    protected $primaryKey = 'id';

    public $incrementing = true;

    public $timestamps = true;

    protected $fillable = ['nip', 'name'];

    public function major()
    {
        return $this->belongsTo('Simta\Models\Major');
    }

    public function projects()
    {
        return $this->belongsToMany('Simta\Models\Project');
    }

    public function user()
    {

```

```

        return $this->morphOne('Simta\Models\User');
    }
    public function meetings()
    {
        return $this->belongsToMany('Simta\Models\Meeting',
        'lecturer_meeting');
    }
}

```

Kode Sumber A.8 Model Lecturer

9. Kode Program Implementasi Model Major

```

<?php
namespace Simta\Models;
use Illuminate\Database\Eloquent\Model;
class Major extends Model
{
    protected $table = 'majors';

    protected $primaryKey = 'id';

    public $incrementing = true;

    public $timestamps = true;

    protected $fillable = ['code', 'name'];

    public function parameters()
    {
        return $this->belongsToMany('Simta\Models\Parameter',
        'major_parameter')->withPivot('value');
    }

    public function workflow()
    {
        return $this->belongsTo('Simta\Models\Workflow');
    }
}

```

Kode Sumber A.9 Model Major

10. Kode Program Implementasi Model Meeting

```

<?php
namespace Simta\Models;

```

```

use Carbon\Carbon;
use Illuminate\Database\Eloquent\Model;

class Meeting extends Model
{
    protected $table = 'meetings';

    protected $primaryKey = 'id';

    public $incrementing = true;

    public $timestamps = true;

    protected $dates = ['held'];

    protected $fillable = ['held'];

    public function __construct()
    {
        $this->held = Carbon::now();
    }

    public function time()
    {
        return $this->session->start . ' - ' . $this->session-
>finish;
    }

    public function session()
    {
        return $this->belongsTo('Simta\Models\MeetingSession');
    }

    public function room()
    {
        return $this->belongsTo('Simta\Models\Room');
    }

    public function project()
    {
        return $this->belongsTo('Simta\Models\Project');
    }

    public function type()
    {
        return $this->belongsTo('Simta\Models\MeetingType');
    }

    public function student()

```

```

    {
        return $this->belongsTo('Simta\Models\Student');
    }

    public function major()
    {
        return $this->belongsTo('Simta\Models\Major');
    }

    public function lecturers()
    {
        return $this->hasMany('Simta\Models\Lecturer',
        'lecturer_meeting');
    }
}

```

Kode Sumber A.10 Model Meeting

11. Kode Program Implementasi Model MeetingSession

```

<?php

namespace Simta\Models;

use Illuminate\Database\Eloquent\Model;

class MeetingSession extends Model
{
    protected $table = 'meeting_sessions';

    protected $primaryKey = 'id';

    public $incrementing = true;

    public $timestamps = true;

    protected $fillable = ['code', 'name', 'start', 'finish'];

    public function major()
    {
        return $this->belongsTo('Simta\Models\Major');
    }
}

```

Kode Sumber A.11 Model MeetingSession

12. Kode Program Implementasi Model MeetingType

```

<?php

namespace Simta\Models;

```



```

use Illuminate\Database\Eloquent\Model;

class MeetingType extends Model
{
    protected $table = 'meeting_types';

    protected $primaryKey = 'id';

    public $incrementing = true;

    public $timestamps = true;

    protected $fillable = ['code', 'name'];

    public function major()
    {
        return $this->belongsTo('Simta\Models\Major');
    }
}

```

Kode Sumber A.12 Model MeetingType

13. Kode Program Implementasi Model Menu

```

<?php

namespace Simta\Models;

use Illuminate\Database\Eloquent\Model;

class Menu extends Model
{
    protected $table = 'menus';

    protected $primaryKey = 'id';

    public $incrementing = true;

    public $timestamps = false;

    protected $fillable = ['name', 'url', 'icon'];

    public function roles()
    {
        return $this->belongsToMany('Simta\Models\Auth\Role',
'menu_role', 'menu_id', 'role_id');
    }

    public function workflows()

```

```

    {
        return $this->belongsToMany('Simta\Models\Workflow',
        'menu_role', 'menu_id', 'workflow_id');
    }

    public function permissions()
    {
        return $this-
>belongsToMany('Simta\Models\Auth\Permission',
        'menu_permission');
    }
}

```

Kode Sumber A.13 Model Menu**14. Kode Program Implementasi Model Parameter**

```

<?php

namespace Simta\Models;

use Illuminate\Database\Eloquent\Model;

class Parameter extends Model
{
    protected $table = 'parameters';

    protected $primaryKey = 'id';

    public $incrementing = true;

    public $timestamps = false;

    protected $fillable = ['code', 'name', 'description'];

    public function majors()
    {
        return $this->belongsToMany('Simta\Models\Major',
        'major_parameter')->withPivot('value');
    }
}

```

Kode Sumber A.14 Model Parameter**15. Kode Program Implementasi Model Progress**

```

<?php

namespace Simta\Models;

```

```

use Illuminate\Database\Eloquent\Model;
use Simta\Models\Traits\ProgresssetStatus;

class Progress extends Model
{
    use ProgresssetStatus;

    protected $table = 'progresses';

    protected $primaryKey = 'id';

    public $incrementing = true;

    public $timestamps = true;

    public $dates = ['event'];

    protected $fillable = ['activity', 'event'];

    const TYPE = 'progress';

    public function toString()
    {
        return $this->activity;
    }

    public function student()
    {
        return $this->belongsTo('Simta\Models\Student');
    }

    public function project()
    {
        return $this->belongsTo('Simta\Models\Project');
    }

    public function status()
    {
        return $this->belongsTo('Simta\Models\ProgressStatus',
'status_id');
    }

    public function major()
    {
        return $this->belongsTo('Simta\Models\Major');
    }
}

```

Kode Sumber A.15 Model Progress

16. Kode Program Implementasi Model ProgressStatus

```
<?php

namespace Simta\Models;

use Illuminate\Database\Eloquent\Model;

class ProgressStatus extends Model
{
    protected $table = 'progress_statuses';

    protected $primaryKey = 'id';

    public $incrementing = true;

    public $timestamps = false;

    protected $fillable = ['code', 'name'];

    public function toHtml()
    {
        if ($this->code == 'belum diverifikasi') {
            return '<span title="Belum Diverifikasi" class="fa fa-question-circle text-default"></span>';
        } elseif ($this->code == 'terverifikasi') {
            return '<span title="Terverifikasi" class="fa fa-check text-success"></span>';
        } elseif ($this->code == 'tidak valid') {
            return '<span title="Tidak Valid" class="fa fa-remove text-warning"></span>';
        }
    }
}
```

Kode Sumber A.16 Model ProgressStatus

17. Kode Program Implementasi Model Project

```
<?php

namespace Simta\Models;

use Illuminate\Database\Eloquent\Model;

class Project extends Model
{
    protected $table = 'projects';
```

```

        protected $primaryKey = 'id';

        public $incrementing = true;

        public $timestamps = true;

        protected $fillable = ['title', 'start', 'file',
'proposal_grade', 'final_grade', 'average_grade'];

        protected $dates = ['start'];

        public function lecturers()
        {
            return $this->belongsToMany('Simta\Models\Lecturer');
        }

        public function student()
        {
            return $this->belongsTo('Simta\Models\Student');
        }

        public function interestArea()
        {
            return $this->belongsTo('Simta\Models\InterestArea');
        }

        public function progresses()
        {
            return $this->hasMany('Simta\Models\Progress');
        }

        public function bookProgresses()
        {
            return $this->hasMany('Simta\Models\BookProgress');
        }

        public function major()
        {
            return $this->belongsTo('Simta\Models\Major');
        }
    }

```

Kode Sumber A.17 Model

18. Kode Program Implementasi Model Proposal

```

<?php

namespace Simta\Models;

```

```

use Illuminate\Database\Eloquent\Model;

class Proposal extends Model
{
    protected $table = 'proposals';

    protected $primaryKey = 'id';

    public $incrementing = true;

    public $timestamps = true;

    protected $fillable = ['title', 'file'];

    public function student()
    {
        return $this->belongsTo('Simta\Models\Student');
    }

    public function project()
    {
        return $this->belongsTo('Simta\Models\Project');
    }

    public function major()
    {
        return $this->belongsTo('Simta\Models\Major');
    }
}

```

Kode Sumber A.18 Model Proposal

19. Kode Program Implementasi Model Room

```

<?php

namespace Simta\Models;

use Illuminate\Database\Eloquent\Model;

class Room extends Model
{
    protected $table = 'rooms';

    protected $primaryKey = 'id';

    public $incrementing = true;

    public $timestamps = true;
}

```

```

        protected $fillable = ['code', 'name'];

        public function major()
        {
            return $this->belongsTo('Simta\Models\Major');
        }
    }
}

```

Kode Sumber A.19 Model Room

20. Kode Program Implementasi Model Student

```

<?php

namespace Simta\Models;

use Illuminate\Database\Eloquent\Model;

class Student extends Model
{
    protected $table = 'students';

    protected $primaryKey = 'id';

    public $incrementing = true;

    public $timestamps = true;

    protected $fillable = ['nrp', 'name'];

    public function major()
    {
        return $this->belongsTo('Simta\Models\Major');
    }

    public function educationDegree()
    {
        return $this->belongsTo('Simta\Models\EducationDegree');
    }

    public function degreeType()
    {
        return $this->belongsTo('Simta\Models\DegreeType');
    }

    public function projects()
    {
        return $this->hasMany('Simta\Models\Project');
    }
}

```

```

    public function user()
    {
        return $this->morphOne('Simta\Models\User');
    }

    public function project()
    {
        return $this->belongsTo('Simta\Models\Project');
    }

    public function progresses()
    {
        return $this->hasMany('Simta\Models\Progress');
    }

    public function bookProgresses()
    {
        return $this->hasMany('Simta\Models\BookProgress');
    }

    public function meetings()
    {
        return $this->hasMany('Simta\Models\Meeting');
    }
}

```

Kode Sumber A.20 Model Student

21. Kode Program Implementasi Model User

```

<?php

namespace Simta\Models;

use Illuminate\Auth\Authenticatable;
use Illuminate\Auth\Passwords\CanResetPassword;
use Illuminate\Contracts\Auth\Authenticatable as
AuthenticatableContract;
use Illuminate\Contracts\Auth\CanResetPassword as
CanResetPasswordContract;
use Illuminate\Database\Eloquent\Model;
use Zizaco\Entrust\Traits\EntrustUserTrait;

class User extends Model implements AuthenticatableContract,
CanResetPasswordContract
{
    use Authenticatable, CanResetPassword, EntrustUserTrait;

    /**

```



```

    * The database table used by the model.
    *
    * @var string
    */
protected $table = 'users';

/**
 * The attributes that are mass assignable.
 *
 * @var array
 */
protected $fillable = ['name', 'username', 'password'];

/**
 * The attributes excluded from the model's JSON form.
 *
 * @var array
 */
protected $hidden = ['password', 'remember_token'];

public function role()
{
    return $this->belongsTo('Simta\Models\Auth\Role');
}

public function roles()
{
    return $this->belongsToMany('Simta\Models\Auth\Role');
}

public function inRole($name)
{
    return $this->role->name == $name;
}

public function person()
{
    return $this->morphTo();
}

public function major()
{
    return $this->belongsTo('Simta\Models\Major');
}

public function menus()
{
    return $this->role->menus;
}

```

```
}

```

Kode Sumber A.21 Model User

22. Kode Program Implementasi Model Workflow

```
<?php
namespace Simta\Models;

use Illuminate\Database\Eloquent\Model;

class Workflow extends Model
{
    protected $table = 'workflows';

    protected $primaryKey = 'id';

    public $incrementing = true;

    public $timestamps = false;

    protected $fillable = ['name', 'description'];

    public function menus()
    {
        return $this->belongsToMany('Simta\Models\Menu',
            'menu_role', 'workflow_id', 'menu_id')->withPivot('role_id');
    }

    public function permissions()
    {
        return $this-
>belongsToMany('Simta\Models\Auth\Permission',
            'permission_role', 'workflow_id', 'permission_id')-
>withPivot('role_id');
    }
}
```

Kode Sumber A.22 Model Workflow

23. Kode Program Implementasi Model Permission

```
<?php namespace Simta\Models\Auth;

use Zizaco\Entrust\EntrustPermission;

class Permission extends EntrustPermission
{

```

```

        protected $fillable = ['name', 'display_name',
        'description'];

        public function workflows()
        {
            return $this->belongsToMany('Simta\Models\Workflow',
            'permission_role', 'permission_id', 'workflow_id');
        }

        public function menu()
        {
            return $this->belongsToMany('Simta\Models\Menu',
            'menu_permission');
        }
    }

```

Kode Sumber A.23 Model Permission

24. Kode Program Implementasi Model Role

```

<?php namespace Simta\Models\Auth;

use Zizaco\Entrust\EntrustRole;

class Role extends EntrustRole
{
    protected $fillable = ['name', 'display_name',
    'description'];

    public function menus()
    {
        return $this->belongsToMany('Simta\Models\Menu',
        'menu_role', 'role_id', 'menu_id')->withPivot('workflow_id');
    }

    public function workflows()
    {
        return $this->belongsToMany('Simta\Models\Workflow',
        'menu_role', 'role_id', 'workflow_id')->withPivot('menu_id');
    }
}

```

Kode Sumber A.24 Model Role

[Halaman ini sengaja dikosongkan]

DAFTAR PUSTAKA

- [1] Tim Penyusun Katalog Teknik Informatika, Katalog Program Studi S1 & S2 Teknik Informatika, Surabaya, 2010.
- [2] Menteri Pendidikan Dan Kebudayaan Republik Indonesia, *Peraturan Menteri Pendidikan Dan Kebudayaan Republik Indonesia Nomor 86 Tahun 2013 Tentang Organisasi Dan Tata Kerja Institut Teknologi Sepuluh Nopember*, 2013.
- [3] Microsoft Developer Network, "Chapter 3: Workflow and Process," Microsoft, [Online]. Available: <http://msdn.microsoft.com/en-us/library/bb833024.aspx>. [Accessed 29 Desember 2014].
- [4] A. Ailamaki, Y. E. Ioannidis and M. Livny, "Scientific Workflow Management by Database Management".
- [5] D. O'Sullivan and R. Richardson, "Legacy Systems Migration - A Method and its Tool-kit Framework," p. 1, 1997.
- [6] T. Otwell, "Laravel Philosophy," [Online]. Available: <http://laravel.com/docs/4.2/introduction/#laravel-philosophy>. [Accessed 29 Desember 2014].

[Halaman ini sengaja dikosongkan]

BIODATA PENULIS



Penulis, Ifan Iqbal, lahir di Kabupaten Bangkalan pada tanggal 29 September 1991. Penulis adalah anak kedua dari tiga bersaudara yang tinggal dan dibesarkan di Bangkalan.

Penulis menempuh pendidikan formal di SDN Kemayoran 1 (lulus 2004), SMPN 2 Bangkalan (2004-2007), SMAN 1 Bangkalan (2007-2010). Pada tahun 2011, penulis memulai pendidikan S1 Jurusan Teknik Informatika, Fakultas Teknologi Informasi di Institut Teknologi Sepuluh Nopember, Surabaya, Jawa Timur.

Di Jurusan Teknik Informatika, penulis mengambil bidang minat Rekayasa Perangkat Lunak. Beberapa bidang yang menarik bagi penulis di antaranya adalah analisis dan perancangan perangkat lunak, manajemen pengembangan perangkat lunak, perancangan perangkat lunak, dan konstruksi perangkat lunak. Sejak tahun 2013, penulis aktif menjalankan kegiatan magang di Lembaga Pengembangan Teknologi Sistem Informasi (LPTSI) ITS. Penulis aktif dalam beberapa forum pengembangan dalam jaringan dengan fokus pembahasan Laravel. Atas kerja sama dengan LPTSI dan PT. Profio Teknova Indonesia, penulis dalam beberapa kali kesempatan pernah menjadi pemateri dalam acara pelatihan kerangka pemrograman Laravel yang diadakan di DILo ITS. Penulis dapat dihubungi melalui alamat email ifaniqbal.com@gmail.com.